CMP
United Business Media

PRODUCT REVIEWS: ZONA AND BUTTERFLY.NET

FEBRUARY 2003

Game Developer

ON THE FRONT LINE OF GAME INNOVATION

The Art and Tech of Insomniac's
Character Animation System

2003 Game Developers Conference PREVIEW

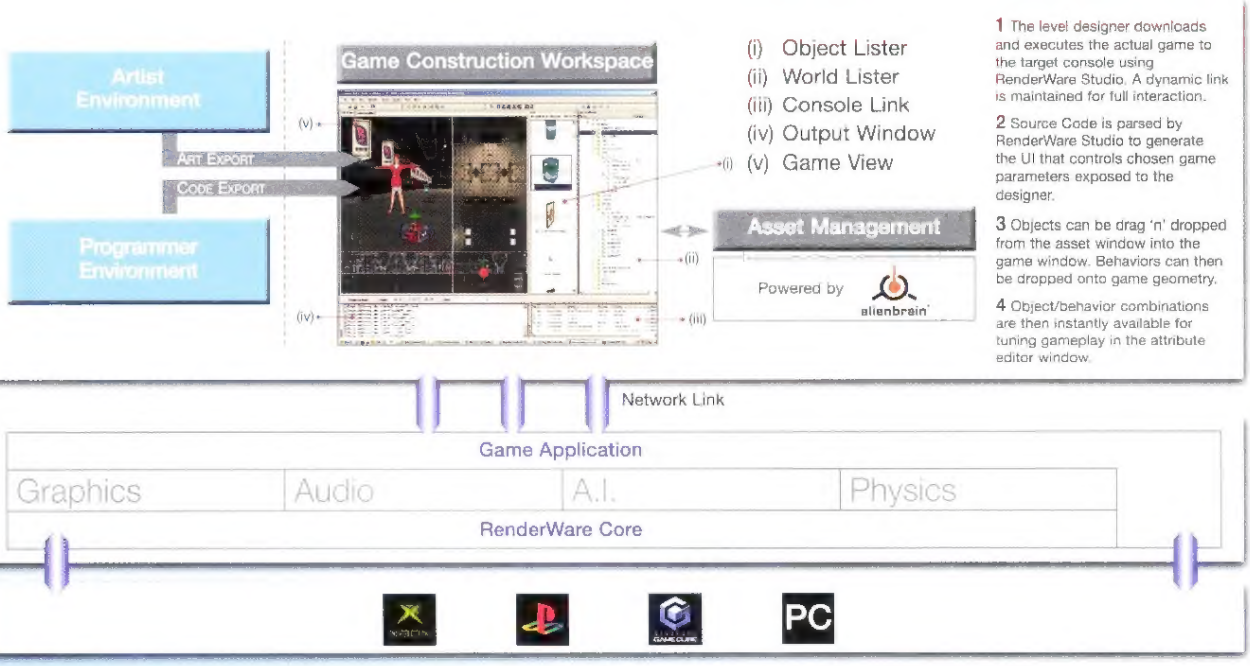
Postmortem: Ensemble Studios' Age of Mythology

The Future of Narrative-Based Games

$5.95US $6.95CAN
02
0 71486 02133 9
Display Until February 28, 2003

RenderWare Studio

>>Powerful, Intuitive, Fast...


Artist Environment
Art Export
Code Export
Programmer Environment
Game Construction Workspace
(i) Object Lister
(ii) World Lister
(iii) Console Link
(iv) Output Window
(v) Game View
Asset Management
Powered by alienbrain
1 The level designer downloads and executes the actual game to the target console using RenderWare Studio. A dynamic link is maintained for full interaction.
2 Source Code is parsed by RenderWare Studio to generate the UI that controls chosen game parameters exposed to the designer.
3 Objects can be drag 'n' dropped from the asset window into the game window. Behaviors can then be dropped onto game geometry.
4 Object/behavior combinations are then instantly available for tuning gameplay in the attribute editor window.
Network Link
Game Application
Graphics
Audio
A.I.
Physics
RenderWare Core
PC

RenderWare Studio is a unique collaborative game production system that allows the whole team to create, view and tune games simultaneously in real-time on multiple platforms.

RenderWare Studio brings together all the essential tools and technologies (graphics, physics, audio, asset management, exporters, viewers) into a single unified production system and generates an intuitive UI that controls all game attributes. Entire development teams and their publishers are provided with a real-time link to all target platforms, ensuring a live build of the game is immediately available.

Encompassing the whole development process from pre-production to QA, the collaborative nature of RenderWare Studio enables all team members, even those in remote locations, to share assets and work in parallel on gameplay, engineering and production in real-time, and remove the time-consuming bottlenecks that can occur.

For further information please visit www.renderware.com, email rw-info@csl.com, or call 512-478-5605

We are proud to be sponsors of...

igda international game developers association

game developers CHOICE awards

We're proud to announce RenderWare is now being used in hundreds of games and would like to congratulate our customers on generating over $ 1 BILLION of revenue from the titles released to date!

Ubi Soft ENTERTAINMENT

"RenderWare Studio will help us to streamline the production of multi-platform games, and maximize the productivity of our developers. It will also allow us to continuously enhance the creative and quality aspects of our games."

Michel Pierfitte, Director of Operations,
Worldwide Production Studios, Ubi Soft Entertainment

namco

"We decided to use RenderWare Studio because of its ability to update the game in real-time and to be able to start building levels immediately was a big plus."

Carl Mey, Technical Director,
Namco Inc.

Sammy Studios

"RenderWare Studio has been an immense tool for us, helping manage the production process so that a greater amount of resources are freed up to focus on creative elements and gameplay. People who were unconvinced of the system's potential as a production enabler pretty much came around when we had a playable prototype level of our games up and running a week after installing it."

Dave Wagner, Executive Technical Director,
Sammy Studios

>> Get To The *Game*

Criterion SOFTWARE

© 2003 Criterion Software Inc. All rights reserved. Criterion and RenderWare are registered trademarks of Canon Inc. PlayStation is a registered trademark of Sony Computer Entertainment Inc. Microsoft is a registered trademark and Xbox is a trademark of Microsoft Corporation. Nintendo is a registered trademark and NINTENDO GAMECUBE is a trademark of Nintendo Co., Ltd. All other trademarks mentioned herein are the property of their respective companies.

CONTENTS

VOLUME 10, NUMBER 2

February 2003

FEATURES

30 Giving Life to RATCHET & CLANK: Complex Character Animations

Staying true to Insomniac's vision of RATCHET & CLANK meant that the digital actors needed to become more than mere cycling automatons; the characters needed to blend physically into their environments, emotionally into their situations, and expressively into the game's narrative.

John Lally

38 Inside the 2003 Game Developers Conference

GDC is right around the corner. Find out what some of this year's speakers have planned for attendees and what they think about another year of GDC.

Jennifer Olsen

44 Postmortem: Ensemble Studios' AGE OF MYTHOLOGY

Heroes and gods. The stuff of myths also happens to be the main ingredient in Ensemble Studios' newest installment in the AGE OF series. How well did the team build on a classic game system while still incorporating fresh gameplay ideas and technologies?

Ian Fischer and Greg Street

DEPARTMENTS

4 GAME PLAN *Jennifer Olsen*
Where's the Party?

6 INDUSTRY WATCH *Everard Strong*
Microsoft spends, Activision deals, Take-Two snags an Angel, and more

9 PRODUCT REVIEWS
Massively Multiplayer Online Game Middleware

14 PROFILES *Everard Strong*
Matt Toschlog's descent into Outrage

COLUMNS

16 THE INNER PRODUCT *Jonathan Blow*
Clustering

22 ARTIST'S VIEW *Hayden Duvall*
What's Wrong with Our Games? Part 2

26 SOUND PRINCIPLES *Simon Amarasingham*
Music That Won't Give You a Rash

28 BETTER BY DESIGN *Noah Falstein*
Good Points Don't Go Down

64 SOAPBOX *Heather Kelley*
Narrative Games: Finding Another Side to the Story

COVER: *Composite of AGE OF MYTHOLOGY's Thor, Isis, and Anubis created exclusively for* Game Developer *by Ensemble Studios' David Cherry.*

Game Developer (ISSN 1073-922X) is published monthly by CMP Media LLC, 600 Harrison St., San Francisco, CA 94107, (415) 947-6000. Please direct advertising and editorial inquiries to this address. Canadian Registered for GST as CMP Media LLC, GST No. R13288078, Customer No. 2116057, Agreement No. 40011901. **SUBSCRIPTION RATES:** Subscription rate for the U.S. is $49.95 for twelve issues. Countries outside the U.S. must be prepaid in U.S. funds drawn on a U.S. bank or via credit card. Canada/Mexico: $69.95; all other countries: $99.95 (issues shipped via air delivery). Periodical postage paid at San Francisco, CA and additional mailing offices. **POSTMASTER:** Send address changes to Game Developer, P.O. Box 1274, Skokie, IL 60076-8274. **CUSTOMER SERVICE:** For subscription orders and changes of address, call toll-free in the U.S. (800) 250-2429 or fax (847) 647-5972. All other countries call (1) (847) 647-5928 or fax (1) (847) 647-5972. Send payments to *Game Developer*, P.O. Box 1274, Skokie, IL 60076-8274. For back issues write to *Game Developer*, 1601 W. 23rd St. Ste. 200, Lawrence, KS 66046-2703. Call toll-free in the U.S./Canada (800) 444-4881 or fax (785) 841-2624. All other countries call (1) (785) 841-1631 or fax (1) (785) 841-2624. Please remember to indicate *Game Developer* on any correspondence.

A New Design Freedom

3Dlabs
Graphics Evolved
A CREATIVE Company

Game Developer FRONTLINE FINALIST '02

Introducing the World's First Fully Programmable Visual Processor

The sky's the only limit to your design creativity with Wildcat® VP—an astounding breakthrough in 3D workstation graphics. The industry's first fully programmable Visual Processing Unit (VPU) powers a family of accelerators that deliver unmatched productivity for professional OpenGL® and Direct3D®—based applications. But that's not all. With unrivaled flexibility, Wildcat VP is enabling a new generation of applications to extend the boundaries of interactive visual realism. Experience the ultimate in graphics design freedom today. Be ready for visual programmability tomorrow. Wildcat VP from 3Dlabs.

Leading OpenGL and Direct3D performance	A full family of 64MB and 128MB boards
Over 200GFlop and 1.2 TeraOp VPU	High performance virtual memory
Professional-grade reliability and quality	Dual independent displays

Wildcat VP

www.3dlabs.com

3Dlabs is a registered trademark of 3Dlabs Inc. in the United States and other countries.

GAME PLAN

LETTER FROM THE EDITOR

Where's the Party?

Now is not the best time in the world to be a third-party game developer. Whereas in the past the natural rhythms of game industry cycles and fortunes mirrored big-bang–big-crunch cycles of consolidation and start-ups, the game industry is now large enough that a long-range forecast of continued consolidation seems more likely.

Going first-party will provide only short-term security for some developers if the next year sees a major shakeout among publishers, as analysts are beginning to predict will happen. Second parties barely exist as such anymore, most of the prominent ones having been fully absorbed or cut loose by their publishers in this console cycle.

Is the industry headed for a shakeout in 2003? If there's one thing to learn from games, it's that very few match-ups don't end up with winners and losers. Consolidation has been a consistent feature of the maturing of most other entertainment media, and currently the odds don't seem good that we will see as many game publishers at the end of 2003 as there are today. Those that go down will take their first-party studios with them, leaving some developers in unfamiliar hands through mergers and acquisitions, and leaving others out in the cold entirely. Developers and publishers in all three major markets — Japan, North America, and Europe — continue to face unique but significant challenges in their respective economies.

The hardware makers are holding — and exerting — an increasing amount of power, making tycoons out of winners and paupers out of losers. On the developer end of the food chain, budgets are being clamped down and plum projects seem to be harder to come by, even for those studios for whom they used to come relatively easy. If you now work for a small studio and the name printed in the corner of your paychecks hasn't changed to that of a larger company in the past couple of years, you could be in for a bumpy ride.

New this month. This issue features the debut of a new editorial property, *Game Developer Mobile*. So much information is swirling around the nascent industry of games for mobile devices that we've settled on a newsletter format, with reporting and analysis from Ben Calica, a veteran game industry writer and analyst. Our goal is to cut through the hype with an independent perspective, while providing the most relevant, up-to-date information and trends in a context that keeps developers' issues and concerns at the forefront. Look for this new feature bimonthly, and be sure to let us know what kinds of news and information you'd like to see in future editions by e-mailing us at editors@gdmag.com.

The next 12 months will be crucial for mobile game developers. In order to carve out a successful role in this market, game developers must provide their clients real value in the form of well-executed games, while asserting their strategic value in a market where most business deals remain ad hoc and few workable standard models have emerged. Handset manufacturers and mobile operators seem convincingly committed to supporting a market for games; let's hope they give developers enough leverage and incentive to make good ones.

Jennifer

Jennifer Olsen
Editor-In-Chief

GameDeveloper
www.gdmag.com
600 Harrison Street, San Francisco, CA 94107 t: 415.947.6000 f: 415.947.6090

Publisher
Jennifer Pahlka *jpahlka@cmp.com*

EDITORIAL
Editor-In-Chief
Jennifer Olsen *jolsen@cmp.com*
Managing Editor
Everard Strong *estrong@cmp.com*
Production Editor
Olga Zundel *ozundel@cmp.com*
Product Review Editor
Daniel Huebner *dan@gamasutra.com*
Art Director
Audrey Welch *awelch@cmp.com*
Editor-At-Large
Chris Hecker *checker@d6.com*
Contributing Editors
Jonathan Blow *jon@number-none.com*
Hayden Duvall *hayden@confounding-factor.com*
Noah Falstein *noah@theinspiracy.com*
Advisory Board
Hal Barwood LucasArts
Ellen Guon Beeman Monolith
Andy Gavin Naughty Dog
Joby Otero Luxoflux
Dave Pottinger Ensemble Studios
George Sanger Big Fat Inc.
Harvey Smith Ion Storm
Paul Steed Independent

ADVERTISING SALES
Director of Sales/Associate Publisher
Michele Sweeney e: *msweeney@cmp.com* t: 415.947.6217
Senior Account Manager, Eastern Region & Europe
Afton Thatcher e: *athatcher@cmp.com* t: 828.350.9392
Account Manager, Northern California & Southeast
Susan Kirby e: *skirby@cmp.com* t: 415.947.6226
Account Manager, Recruitment
Raelene Maiben e: *rmaiben@cmp.com* t: 415.947.6225
Account Manager, Western Region & Asia
Craig Perreault e: *cperreault@cmp.com* t: 415.947.6223
Account Representative
Aaron Murawski e: *amurawski@cmp.com* t: 415.947.6227

ADVERTISING PRODUCTION
Vice President, Manufacturing Bill Amstutz
Advertising Production Coordinator Kevin Chanel
Reprints Cindy Zauss t: 909.698.1780

GAMA NETWORK MARKETING
Director of Marketing Greg Kerwin
Senior MarCom Manager Jennifer McLean
Marketing Coordinator Scott Lyon

CIRCULATION
BPA INTERNATIONAL
Game Developer is BPA approved
Group Circulation Director Catherine Flynn
Circulation Manager Ron Escobar
Circulation Assistant Ian Hay
Newsstand Analyst Pam Santoro

SUBSCRIPTION SERVICES
For information, order questions, and address changes
t: 800.250.2429 or 847.647.5928 f: 847.647.5972
e: *gamedeveloper@halldata.com*

INTERNATIONAL LICENSING INFORMATION
Mario Salinas
t: 650.513.4234 f: 650.513.4482 e: *msalinas@cmp.com*

CMP MEDIA MANAGEMENT
President & CEO Gary Marshall
Executive Vice President & CFO John Day
Chief Operating Officer Steve Weitzner
Chief Information Officer Mike Mikos
President, Technology Solutions Group Robert Faletra
President, Healthcare Group Vicki Masseria
President, Electronics Group Jeff Patterson
President, Specialized Technologies Group Regina Starr Ridley
Senior Vice President, Global Sales & Marketing Bill Howard
Senior Vice President, HR & Communications Leah Landro
Vice President & General Counsel Sandra Grayson
Vice President, Creative Technologies Philip Chapnick

CMP United Business Media
GamaNetwork

WWW.GAMANETWORK.COM

GAME COMPONENTS
GameDeveloper FRONTLINE AWARD '02

Hiring now

**Havok physics for characters.
Not just for bipeds.
Available now.**

Think what you could do with multi-limbed physical ragdolls in your game. See the effects at www.havok.com/havok2

havok

We supply the physics, you supply the imagination.

INDUSTRY WATCH

KEEPING AN EYE ON THE GAME BIZ | everard strong

No more Mr. Nice Box. Microsoft has no plans to ease up on its efforts to carve out a position in the $10 billion videogame market, rather than cutting its losses and exiting from the venture. Industry analysts expect Microsoft to spend more than $2 billion in the next five years on the Xbox. Microsoft's Xbox Live broadband service has seen early success, with a reported 150,000 kits sold in the first week of release.

Metis brands Acclaim. Marc Metis has joined Acclaim Entertainment as its new senior vice president of brand. In his position, he will manage Acclaim's franchises and new brands.

No rust in Activision deal with Stainless Steel. Known for the PC strategy title EMPIRE EARTH, Stainless Steel Studios has signed an exclusive multi-title, multi-year agreement with Activision. Their first title will be another historical RTS game.

NewKidCo posts loss. Due to the delayed release of two titles — LITTLE LEAGUE BASEBALL 2002 (GBA) and TOM & JERRY WAR OF THE WHISKERS (PS2) — NewKidCo announced an operating loss of $2.4 million for Q3 2002, compared to a $607,000 operating loss for the same period in 2001.

EMPIRE EARTH developers Stainless Steel Studios have signed a multi-title deal with Activision.

PC game sale revenue up for 2002, but unit volume down. NPDTechworld revealed that PC game revenues were up 1.2 percent — from $945 million for the first 10 months in 2001 to $956 for that same time period in 2002 — but the number of units sold fell by 6.2 percent, from 44.4 million in 2001 to 41.6 million for those same 10 months in 2002. The research also pointed out that the average selling price of PC games rose from $32 in 2001 to $37 in 2002, but it did not say if this price increase caused the lower volume of unit sales or if there were other, outside factors, such as increased competition from the console industry or other broader economic trends.

Take-Two takes its own Angel. Take Two has acquired Angel Studios, developers of MIDNIGHT CLUB and SMUGGLER'S RUN, for an aggregate $28 million in cash and around 235,000 shares of restricted common stock. Angel Studios will now be known as Rockstar San Diego.

Send all industry and product news to news@gdmag.com.

THE TOOLBOX

DEVELOPMENT SOFTWARE, HARDWARE, AND OTHER STUFF

Calculate game royalties and payouts. PLX Systems introduced a new royalty accounting and licensing system for large game publishers. The product, PLXware Right Track Suite (PLXware/RTS), can be used to process royalties for authors, artists, composers, producers, and other rights-holders. www.plxsystems.com

New tree creation software. IDV recently released SpeedTreeRT, a tree creation tool for game and visual simulation development. The program delivers low-polygon, realistic trees, with adjustable wind effects, seamless LOD transitions, and a library of over 90 trees from 30 core species. www.idvinc.com

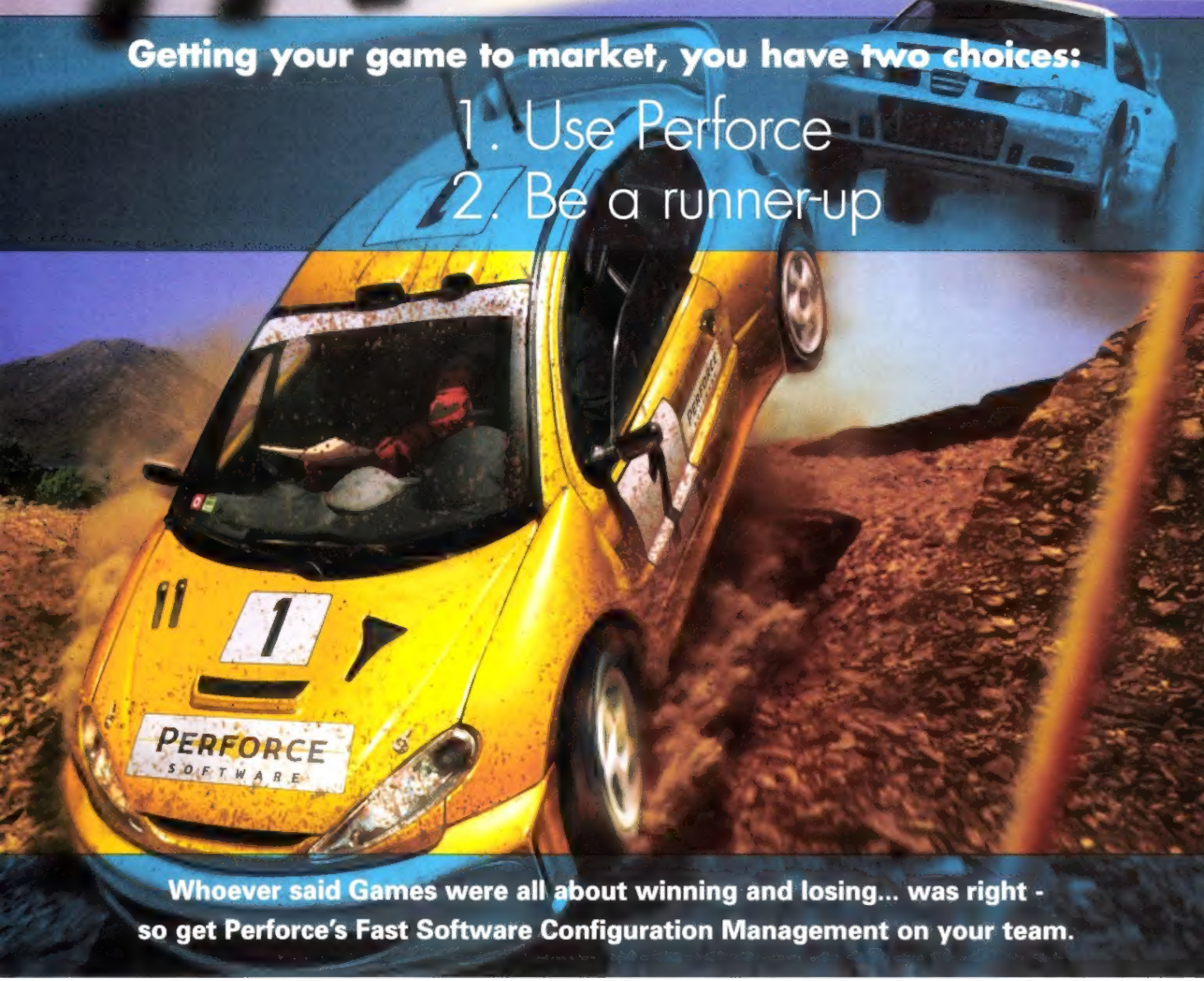
New Intel C++ compiler. Intel's newest C++ compiler, version 7.0, integrates with Microsoft Visual Studio, and the Linux version provides GNU compatibility to C++. The compiler also include an auto-parallelization option that automatically looks for opportunities in applications to create multiple execution threads. www.intel.com

UPCOMING EVENTS CALENDAR

GAME DEVELOPERS CONFERENCE
SAN JOSE CONVENTION CENTER
San Jose, Calif.
March 4–8, 2003
Cost: $150–$1,975
www.gdconf.com

GDC MOBILE
SAN JOSE CONVENTION CENTER
San Jose, Calif.
March 4–5, 2003
Cost: $895
www.gdcmobile.com

MILIA (NEW DATES)
PALAIS DES FESTIVALS
Cannes, France
March 26-28, 2003
Cost: variable
www.milia.com


Getting your game to market, you have two choices:
1. Use Perforce
2. Be a runner-up
1
PERFORCE
SOFTWARE
Whoever said Games were all about winning and losing... was right -
so get Perforce's Fast Software Configuration Management on your team.

Perforce's Software Configuration Management System is the choice of winners, because no other tool can match the speed, control and reliability that it brings to the management of digital assets (binary and text files).

Maintain your top speed no matter how many users or files. At the core of Perforce lies a relational database with well-keyed tables, so simple operations can be accomplished in near-zero time. Larger operations (like labeling a release and branching) are translated into keyed data access, giving Perforce the scalability that big projects require.

Install it fast, learn it fast, execute operations fast. With other SCM systems, developers face an unpleasant choice: do it the right way or do it the fast way. Perforce's speed and reliability mean the fast way is always the right way.

Work anywhere. Perforce is efficient over high-latency networks such as WANs, the Internet and even low-speed dial-up connections. Requiring only TCP/IP, Perforce makes use of a well-tuned streaming message protocol for synchronising client workspace with server repository contents.

Develop and maintain multiple codelines. Perforce Inter-File Branching™ lets you merge new features and apply fixes between codelines. Smart metadata keeps track of your evolving projects even while they develop in parallel.

Supports all development platforms. Perforce runs on more than 50 operating systems, including all flavours of Windows, UNIX, Linux, Mac OS X and more.

Integrate with leading IDEs and defect trackers: Visual Studio.NET®, Visual C++®, Visual Basic®, JBuilder®, CodeWarrior®, TeamTrack®, Bugzilla™, ControlCenter®, DevTrack® packages, and more.

PERFORCE
SOFTWARE

Fast Software Configuration Management *www.perforce.com*

Download your free 2-user, non-expiring, full-featured copy now from www.perforce.com
Free (and friendly) technical support is on hand to answer any and all evaluation questions.

All trademarks used herein are either the trademarks or registered trademarks of their respective owners.

consumer reach

"Next year (in 2003) Nokia expects to ship 50 to 100 million devices that have a color display and an open application development platform. Of these phones, we expect roughly 10 million will be Series 60-based devices."

— Jorma Ollila, Chairman and CEO, Nokia
November 5th 2002

forum.nokia.com/games

+ Download tools and technical documentation on mobile gaming
+ Get complete specifications and stay up to date on new devices
+ Access global distribution channels for mobile games

GDC and GDC Mobile 2003

+ GDC Mobile keynote presentation from Illka Raiskenen, Senior Vice President, Entertainment & Media Business Unit
+ Get your hands on the latest devices and connect with Nokia's games team in the Nokia booth
+ Check out the sponsored sessions during GDC and learn how to build games for N-Gage™ and all of Nokia's platforms

© Nokia 2003

Develop and deploy mobile applications on the world's largest installed base of mobile devices: www.forum.nokia.com/games

NOKIA

SPONSORED BY NOKIA

VOLUME 1 ISSUE 1 FEBRUARY 2003

Game Developer Mobile

Digital Endcaps: The Challenge of Mobile Game Distribution

By Ben Calica

Menuing systems have limitations as a game distribution model

Fueled by the unexpected success of selling ringtones to cell phone users, mobile service operators are opening the doors to an anticipated flood of sales for games. Before grabbing pick and shove to join the gold rush though, game developers need to be aware of some of the challenges in mobile distribution

Challenge #1: Carriers are the gatekeepers.

We should first understand that the developer's first customer isn't the consumer, it's the mobile carriers Physical distribution comes either via a list on the handset that is downloaded over the air or via the carrier s web site. When purchased through the web site, the game is sent through the airwaves by the carriers From a practical standpoint, there is no shareware market for mobile games, and no way to bootstrap. The advantage of this model is that the mechanism for sales is real; it usually ends up being a micropayment added on to the user's phone bill. The downside is that unless the carrier decides to play, the developer is out of luck

Here is where it gets really tough. Both Nokia and Qualcomm have matchmaking services between developers and carriers, through their Tradepoint and BREW services, respectively. But the way it is set up shows their natural bias to serve the carriers "Tradepoint was set up to give developers access to the carriers," says Donna Regenbaum, director of Forum Nokia for the Americas, speaking of their matchmaking service The service allows developers to list their completed products for the carriers

Sun Releases Specs for Next-Generation J2ME APIs

By Ben Calica

Sun Microsystems has released MIDP 2.0, the next-generation specifications for the Mobile nformation Device Profile. This release includes the final MIDP 2.0 spec fication, reference implementation, compatibility test suite, and a beta version of the J2ME Wireless Toolkit 2 0 According to Sun, there are now more than 50 million Java-enabled mobile devices from some 20 manufacturers currently in consumers hands

The specification includes a number of mobile-specific APIs New features in 2.0 include

- Enhanced user interface that provides more common UI elements and better handling for devices with different screen sizes
- Media support to help leverage the more sophisticated audio capabilities of newer handsets
- Game support to provide a standard foundation for building games, and an enhanced

INDUSTRY news

Mforma and Activision Boldly Go...

Seattle-based wireless developer Mforma announced with Activision that they will publish titles based on the license from the new *Star Trek* movie, *Star Trek Nemesis*. The first title will be a space shooter that is set for release at the same time as the movie, and the second will be released with the DVD

www.mforma.com

www.activision.com

Metrowerks Releases Wireless Studio, PDA Edition

Metrowerks has released the Wireless Studio 7, PDA Edition for their CodeWarrior development environment. It provides a debugging enabled version of the Insignia PersonalJava Virtual Machine, which also runs in the BREW environment. It works with and/or includes SDKs from handset vendors including Motorola, iDEN, Sony Ericsson, Siemens, Nokia, Sprint and Sun's

CONTINUED ON PAGE 4 COLUMN 3

inside...

JAMDAT: Bowling for a Perfect Game 2

Nokia N-Gages Gamers, Will Developers Follow? 3

Mobile Developer Resources 4

GameDeveloper
Mobile
VOLUME 1 ISSUE 1
FEBRUARY 2003

EDITOR FOR MOBILE DEVELOPMENT
Ben Calica • bencalica@yahoo.com
ART DIRECTOR
Audrey Welch • awelch@cmp.com

A PUBLICATION OF GameDeveloper
EDITOR-IN-CHIEF
Jennifer Olsen • jolsen@cmp.com
MANAGING EDITOR
Everard Strong • estrong@cmp.com
PRODUCTION EDITOR
Olga Zundel • ozundel@cmp.com
PRODUCT REVIEWS EDITOR
Daniel Huebner • dan@gamasutra.com

PUBLISHED BY THE GAMA NETWORK
A DIVISION OF CMP MEDIA LLC
PUBLISHER
Jennifer Pahlka • jpahlka@cmp.com
DIRECTOR OF SALES
Michele Sweeney • msweeney@cmp.com
415 947-6217

CMP GamaNetwork

to browse and potentially choose They do have both parties agreeing to use standard deal types to smooth negotiations between industries that barely seem to speak the same language, but it's an almost unheard-of situation for a game developer to need to complete a game before they even know if there is potential distribution.

Challenge #2: Digital distribution space is limited.

So why would the carriers be such a barrier? After all, the more games they sell, the more money everybody makes, right? The problems are that there is a limit to the distribution space, and there is a desire to have no support costs The widest distribution is from the nested lists that are pulled directly down onto the mobile phones. But even in the best of circumstances only a few hundred titles could be viewed this way

Given that Nokia claims to have more than 500,000 inquiring mobile developers, that is a pretty big distribution gap. "In future versions we are hoping to have software in BREW that is more intelligent about what types of software we push down to a customer's phone," says Jeremy James, senior director of marketing for Qualcomm Internet Services. "If my usage pattern shows me that I just buy shooters, the shopping selection should reflect that."

Even for the titles that are distributed via the carrier's web site, there are conflicting motivations to carrying too many titles. The prices on the games are so low that one tech support call can wipe out a customer's entire profit margin.

Rather than take the chance, the carrier's natural tendency is to opt for caution. BREW provides a quality assurance by requiring testing before distribution Nokia has a "Nokia OK" testing option, which a developer can pay for, that also ensures games come up at the top in their Tradepoint system. The other factor is that the carriers have learned not to overwhelm their customers by offer-

case study

JAMDAT: Bowling for a Perfect Game

By Ben Calica

JAMDAT BOWLING has become something of a poster child for mobile gaming. It's hard to find a next-generation handset that doesn't have it as a download option, and many initial ads pushing mobile phone gaming have featured it. But bringing this seemingly simple, single-level bowling game to market presented a number of challenges, while helping JAMDAT acquire some of the technical and business expertise that is becoming their stock in trade.

With investments from both Qualcomm and Sun, and distribution deals with most major carriers, JAMDAT looks well positioned in the mobile game business As a company, JAMDAT adopted the developer/publisher model in the style of Activision, which is no surprise, since a number of their key executives have their roots in the "traditional" videogame publishing powerhouse.

To make JAMDAT BOWLING, the company worked with Michael Cartabiano at Sennari Games as an external development house. Working with an in-house producer, JAMDAT brought the game concept, and the two companies planned out the gameplay together.

"We chose bowling because we thought it would work on the Sharp Z800," says JAMDAT president Scott Lehman, speaking about the limited-power handset that was the company's initial target. "We adopted the model of making the lead SKU great and expanding out from there. As we started working on the game, we realized we needed to make the physics great, but we were seriously limited on the processing power the phone had to offer."

Mike and his team at Sennari used every trick at their disposal to create a physics model that would work on the Z800 handset. "That's when the real fun began," remarked Lehman "Every handset is like developing for a new platform. We had to redo the physics model for each one. It's like being in the PC days not only prior to DirectX, but before there were even standards for sound cards."

They used BREW for the first version of BOWLING. "Both BREW and J2ME are fine for doing development on these phones. The ability to deploy on lots of different handsets isn't really relevant, since you need to tweak for each one anyway," said Lehman. "BREW will probably feel more familiar to hardcore game developers, since it uses C to access other APIs and ends up as compiled code. But there are a ton of Java developers these days, and they can do great work in J2ME. And the next version of J2ME is adding direct support for a lot more primitives, so it continues to get easier to work with."

Despite the inherent restrictiveness, the development environment doesn't end up being the really tough part of the problem, according to JAMDAT CEO Mitch Lasky "The challenge with wireless is stripping down to what do you really need. It has to be fun in five minutes. The key is finding the essence of the game."

Lehman adds laughingly of people who think the simpler game designs are easier, "It's easy like writing the perfect haiku is easy." The need for deft execution on simple game designs has brought about a resurgence in the value of game industry oldtimers. "These guys know what it's like to write with very limited capabilities and assets. They are incredibly valuable," comments Lasky.

JAMDAT BOWLING took them longer than they expected, but still ended up much shorter than a standard game development cycle. "We can get a game out in anywhere from six to 18 months,

ing them too many options and causing decision paralysis Ultimately it is up the individual carrier what they decide to do Japan's DoCoMo, for example, has opted to provide access to thousands of applications

Challenge #3
Pricing models are all experimental.

The final challenge is that the pricing models for the games remain experimental "We're still very early in the evolution of the mobile gaming market. I don't know what is going to work yet," says Nokia's Regenbaum One initial model being tested is a limited free trial, with either a monthly rental fee in the $1 to $2 range or the option to buy an unlimited license for $4 to $7

Such pricing schemes show how early in the game it is. Currently, when a customer buys an unlimited license, it is for the device, not for the phone number or the person If the game sales are successful, that limitation may result in people being less likely to upgrade to new handsets after they invest in a lot of software, something that neither the handset makers nor carriers want

The good news overall is that people are buying. 'We won't give our exact numbers, but let's just say that if our business plans called for X sales to have a profitable title, we're seeing between five and nine times X,' says Mitch Lasky, CEO of mobile game publisher JAMDAT [see Case Study, JAMDAT Bowling for a Perfect Game,' below]

Nokia N-Gages Gamers, Will Developers Follow?

By Ben Joyce

After a tantalizing announcement in November, Nokia is planning a more formal launch of their new dedicated gaming handset, N-Gage, in February

The handset features a more traditional game pad layout and is built on the Symbian operating system and the Nokia Series 60 platform. Games can be delivered over the air or via MMC cards

The N-Gage also features both Bluetooth for local multiplayer gaming and over-the-air connection for wider multiplayer gaming, which Nokia points to as distinguishing characteristics of the platform Nokia is interested in the N-Gage both as a device to sell to consumers and as a Trojan horse to bring game developers to the Series 60 platform Large companies such as Sega are already working on titles for the launch and Electronic Arts is in negotiations with Nokia to support the device

depending on the complexity of the game," says Lasky. "The interesting thing is that the asset creation isn't the bottleneck, it's the gameplay." Because the asset space is so limited, it is easy to change the content significantly, even late in the process "Bowling is one level, but it took us 25 passes to get the interface and gameplay right. It was very satisfying to spend the time that way," he says

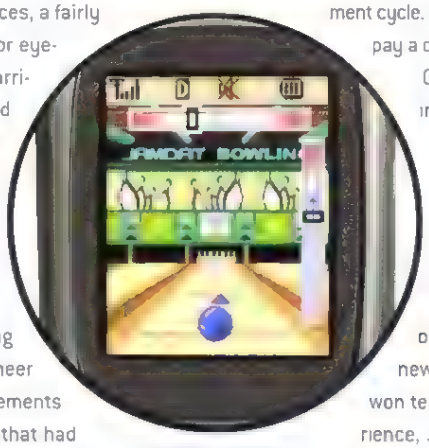
Once they got the first version done, the problems shifted from technical to business. "Two years ago when we got into the business, we were unusual because we were game people. Everyone else was from the wireless industry trying to make games," remembers Lasky. Predictably, given the combination of a limited platform and people who weren't born and bred to the game, the first titles were less than perfect. "There is no place in this business for shovelware," asserts Lasky

The company's founders happened to predict that there would be a market for people buying games on mobile devices, a fairly unpopular notion in the "give it away for free for eyeballs" days of the mid-boom Postulating that carriers might only be able to handle a few hundred games at a time, Lasky recalls, they "decided that there would be a space for a publisher, in the sense that they would fund good games and bring them to the carriers. The carriers have limited the number of publishers that they are going to deal with to five or six "

JAMDAT also faced the problem of deciding how many handsets to develop for, given the sheer variety. "When we started, there were no requirements but the carriers that we support all the devices that had next-generation capability We opted to develop for all the devices we could, because the incremental cost of developing for them was more than covered by the sales we were getting," remarked Lasky "Next year will be different. Verizon is starting to require support of a large number of their next-generation handsets to include a game in its stable."

So far, internal expertise has been one of their big strengths as a publisher "We haven't found any good external resources yet for the nuances for all the handsets," laments Lasky. "The people we have here that have learned about them over the last two years are worth their weight in gold "

In addition to in-house projects, JAMDAT is looking to fund game developers to build more games in the traditional game-publishing model. They are also looking to established developers to add to their stable of products "The teams usually needs to be a designer, artist, and programmer, and the projects can be done in six months, 90 days for a prototype," Lasky says "This is a cool thing to do with teams in between a standard PS2 development cycle. We think the big developers will slip them in to help pay a couple of salaries during that down time."



JAMDAT's ubiquitous bowling game has proved popular

Overall, JAMDAT remains optimistic about this industry's future, a rarity in these gloomy economic times. "There are about 137 million handsets out there that get replaced every 18 months or so. Of those, only about 4 million are next generation, and half of those are Nextel, who are using them for business only That means we should have at least three years of hyper-accelerated growth as people upgrade to new handsets," predicts Lasky Given their hard-won technical expertise and game and business experience, JAMDAT is clearly positioning themselves to be one of the big Kahunas on that wave

www.jamdat.com

set of graphic capabilities to simplify game implementation and to address some of the issues that arose in the first J2ME

- Support for expanded connectivity for standards beyond HTTP, such as server sockets and serial port communications.
- Push Architecture to provide a server push model where MIDlets can be activated when a device receives data from a server.
- OTA (over the air) provisioning to assist in deploying and setting products over the air in a standard way
- End-to-end security that supports HTTPS, SSL, and WTLS protocols to enable transmission of encrypted information.

More information:

J2ME: http://java.sun.com/j2me

MIDP: http://java.sun.com/products/midp

MIDP datasheet: http://java.sun.com/products/midp/midp-ds.pdf

next issue

- Upgrading your favorite Nokia products
- So Many Handsets, So Little Code
- PDA vs. Phone OS: Who Will Win the Gamers' Hearts?

mobile developer

BREW

Qualcomm's site for developer forums, developer lab, BREW SDK, and developer directory.

www.qualcomm.com/brew
Qualcomm Inc.
5775 Morehouse Drive
San Diego, CA 92121
(858) 587-1121

Forum Nokia

Nokia developer support: SDKs, J2ME developer information, Tradepoint marketplace for selling games to carriers.

www.forum.nokia.com

Java 2, Micro Edition (J2ME)

One of two main standards for mobile phone content development

http://java.sun.com/j2me
Sun Microsystems
4150 Network Circle
Santa Clara, CA 95054
(800) 555-9SUN or (650) 960-1300

Microsoft Smartphone 2002

Microsoft's phone-based OS

www.microsoft.com/mobile/smartphone

The Mobile Games Interoperability Forum (MGIF)

Founded by Ericsson, Motorola, Nokia, and Siemens. Established to define mobile game interoperability specifications and APIs.

www.mgif.org
MGIF Administrative Team
mgif-info@mgif.org
(732) 465-6486

Open Mobile Alliance (OMA)

Standards group for the wireless industry, now teamed with the MGIF

www.openmobilealliance.org
Open Mobile Alliance Ltd
2570 W. El Camino Real, Suite 304
Mountain View, CA 94040
info@mail.openmobilealliance.org
(650) 949-6760

Symbian

Operating system for wireless phones.

www.symbian.com
Symbian Inc.
915 Middlefield Road, Suite 4
Redwood City, CA 94063
(650) 365-6300

INDUSTRY news

CONTINUED FROM PAGE 1

Wireless Toolkit for creating Java 2 Micro Edition (J2ME) applications and games. This version adds support for Pocket PC and the Sharp Zaurus. It retails for $599 and joins their products for the Symbian OS, Palm, and ARM.

www.metrowerks.com

JAMDAT Completes the Quest

JAMDAT Mobile acquired rights from New Line Cinema to develop and publish wireless content based on the entire *Lord of the Rings* movie series. Under the terms of the multi-year deal, JAMDAT Mobile will publish downloadable content based on *The Lord of the Rings: The Two Towers* and the third installment of the trilogy, *The Return of the King*. This adds to JAMDAT's currently available *Lord of the Rings* titles.

www.jamdat.com

Brewing Java over the Air

Qualcomm has announced that Insignia's Mobile Foundation J2ME virtual machine for BREW can now be provisioned automatically over the air to a BREW-enabled handset. Through this mechanism, carriers who previously only provided BREW-developed games can add the capability to sell J2ME games to their existing customer base entirely via wireless download. This move is seen as a way to bridge the gap between the competing standards, although each level of abstraction will have performance consequences for certain types of games.

www.qualcomm.com/brew

EverQuest in Hand?

Sony Online Entertainment is slated to bring the world of EverQuest to wireless handsets, with EverQuest: Hero's Call scheduled for release in January 2003. This will be a single-player game with 32 dungeons and access to more than 120 spells. The spells and weapons will be familiar to EverQuest players, although the single-user experience will be strange for a game whose main claim to fame is its massively multiplayer capability. The first version was built using BREW.

www.sonyonline.com

Free Fishing Game for Free Beer

Buzzards Bay Brewing has released a Wireless Fishing game developed by Yellow Pepper Inc., a mobile marketing company. Users go to www.buzzardsbrew.com and enter their phone number to have the game sent directly to their handset. The winner will win a year's supply of the company's beer. The game will only operate with two-way SMS-enabled phones on Verizon, AT&T, Cingular, and Voicestream Networks.

www.buzzardsbrew.com
www.yellowpepper.com

New Worlds, New Battlefields: Massively Multiplayer Online Game Middleware

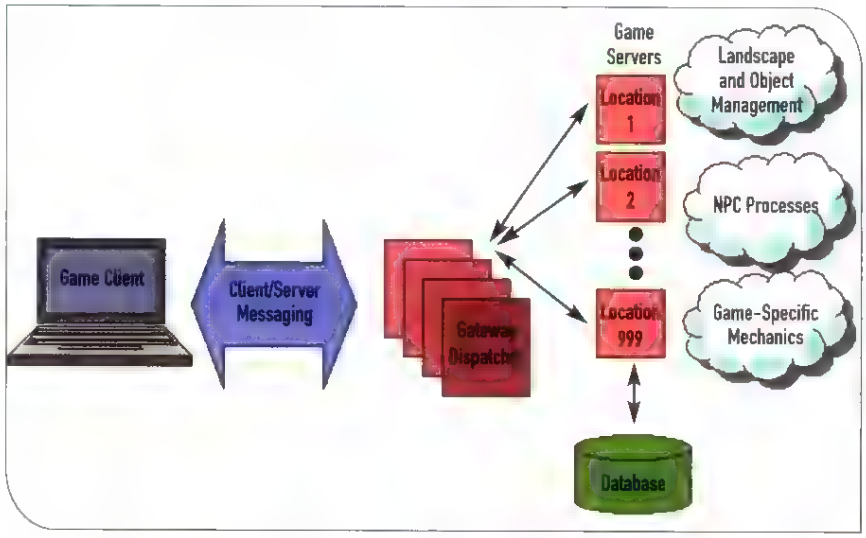
by mitch ferguson and michael ballbach

Creating a massively multiplayer online game is game development's equivalent of a moon shot. It's expensive, technically difficult, and can take years to complete, and yet everyone wants to give it a try.

The technology required is not easy for newcomers to develop. Sensing the need for software that handles a variety of functions, including billing, patching, support, administration, client messaging, stable servers, data persistence, and server clusters, several companies are developing middleware libraries and products to ease these hurdles.

Butterfly.net and Zona's Terazona are two products that provide nearly complete MMOG solutions. Both supply the skeleton of an MMOG, leaving only the skin of a graphics engine, a billing system, and the guts of the actual game mechanics to be defined.

Taken at the highest level, Butterfly.net and Terazona use the same solution strategy. The details are different enough that the environment and personality of the developer will clearly suggest one over the other. Their highest-level strategy is not appropriate for all future MMOGs, however, and whether you would use Butterfly.net or Terazona depends on if your game fits into their mold.



A simplified version of Butterfly.net and Terazona architecture

Common Parts of the High-Level Design

Messaging to dispatcher or gateway. Both systems have a simple API to connect, validate an account, get available game characters, and connect with a particular character. This API uses a client-server messaging system, which is easily extensible, reliable, and simple to integrate into a game client.

Game servers and their landscape. The developer divides up their world into fixed sections, with the size of each section dependent on how much action is expected to occur there. The actor/object positioning and landscape collision for each section is simulated on a single server. There can be multiple sections on a single server, but there cannot be multiple servers for a single section. These section servers are the final destination of the game client messages. The boundaries between these world locations are implemented in different ways for Butterfly.net and Terazona, but both make information about things on the other sides of these boundaries available to the game client.

These servers are divided into two separate halves: the generic landscape functionality and the game-specific mechanics. The developer writes the game-specific mechanics to take care of the simple rules such as trading, grouping, and targeting. In addition, the devel-

MITCH FERGUSON | *Mitch was a lead engineer on Maxis's The Sims Online and Jaleco's Lost Continents, and now works for a MMO startup in San Francisco called Perpetual Entertainment.*

MICHAEL BALLBACH | *Mike has been a server engineer at VR1/Jaleco for six years, was the lead server engineer on Jaleco's Lost Continents, and now works for Perpetual Entertainment.*

oper must write a separate process that simulates monsters.

Game-specific mechanics. Messages from the client are forwarded to the game-specific functionality in case special physics or movement rules are in play. Butterfly.net has you write this code in Python, which is interpreted by the built-in interpreter. Terazona asks that you link in a shared library to receive these callbacks. Either way, this rule-checking code is used to validate game state and client messages.

NPCs and monsters. When it comes to monster AI, both systems pass the buck. Terazona has NPC servers connect as a privileged game client. Butterfly.net has a separate NPC server for every section. In both cases the developer is in charge of how these separate servers are designed and coded. There is an API for receiving callbacks and sending messages to the landscape and the game clients.

Persistence. The player data structure (or parts of it) is automatically persisted by both systems. Butterfly.net requires that you define all the state variables that are needed (also in the database), and it persists the player's data in the clear when it can. Terazona saves out the player data as a block into the database. This saves you from having to define a table, but it also prevents you from performing any simple queries on the player data.

Games that work with these assumptions. While both systems claim to be very extensible, in fact both vendors are open to deals that include the source. It should then be possible to shoehorn something unique into either of these frameworks, though this could offset the original reason for using one of these middleware solutions in the first place.

Most of the last generation of massively multiplayer RPGs would have been able to use either of these systems. Their landscapes are stable, the population density of various areas is easily predicted, and the NPC load isn't outrageous. Both systems allow the tuning of the performances characteristics to allow for a faster, twitch style of game.

However, many of the coming generation of MMOGs are starting to break the old mold. THE SIMS ONLINE and TABULA RASA both will have a very dynamic system of landscapes and server processes coming in and out of existence. Also, if someone decides to try a game with a random landscape system or an often modified and persisted landscape, they might have difficulty.

The Details

Terazona. Terazona server components are implemented almost entirely in Java. A lot of components make up the server suite, including the authentication server; these include the administration component, the NPC servers, the GSS (or Game State Servers), and others. Intercomponent communication is realized by an implementation of the Java Message-oriented middleware standard provided by ICE Technology.

The Game State Servers require the developer to implement a set of functions (their GSAPI, Game State API) in C/C++ and compile them into a DLL or shared

Selecting Networking Middleware

When considering networking middleware, things aren't as straightforward as you may think. Networking is not just a numbers game; you can't look at fill rate and rendering features as you might graphics engines. Networking is more closely tied with players rather than the game's features. It's what allows your players to play against each other, and unlike the game, your players aren't such a known or predictable quantity.

— Crosbie Fitch

SELECTED PROVIDERS:

TERAZONA
See page 12 for contact details

BUTTERFLY.NET
See page 12 for contact details

OPEN SKIES
Cybernet Systems
Ann Arbor, Mich.
www.openskies.net

VAST
Rebel Arts
Calabasas, Calif.
info@rebelarts.com
www.rebelarts.com

TERRAPLAY
Terraplay AB
Solna, Sweden
+46 (8) 764 91 00
www.terraplay.com

DEX
Horizon, a Glimpse of Tomorrow
Monrovia, Calif.
(626) 446-8925
www.horizongot.com

NET-Z AND ETERNA
Quazal
Montreal, Quebec, Canada
(514) 395-4646
www.quazal.com

SIMINTERNET
MÄK Technologies
Cambridge, Mass.
(617) 876-8085
www.mak.com

LITHTECH DISCOVERY SYSTEM
Lithtech
Kirkland, Wash.
(425) 739-1659
www.lithtech.com

TURBINE ENGINE
Turbine Entertainment Software
Westwood, Mass.
(781) 407-4000
www.turbinegames.com

NEL
Nevrax
London, U.K.
business@nevrax.com
www.nevrax.com

TWISTED
Twisted Matrix Labs
www.twistedmatrix.com

Who:	Matt Sivertson
What:	Wheel of Fortune & Jeopardy
Where:	This is wireless – there is no "where"
Why:	The future always arrives earlier than you'd expect

Matt Sivertson is a developer working for Sony Online Entertainment. He's the brains behind both Wheel of Fortune and Jeopardy – two new BREW™ applications wireless users can download and play whenever they want. Like Sivertson, more and more developers are recognizing the rewards of the BREW solution. "There's a huge untapped market out there," he says. "And BREW has the carrier support to get my technology into the hands of those people." Commercial services are launched and BREW applications are hitting the market. And they're hitting the market now. Games and entertainment, messaging and email, news, weather, sports, stock trades, position location, ringers... you name it. If you aren't developing for BREW, you aren't developing to your potential. To get started, visit www.qualcomm.com/brew/gd.

brew™

Customize. Personalize. Realize.™

©2003 QUALCOMM Incorporated. All rights reserved. QUALCOMM is a registered trademark of QUALCOMM Incorporated. BREW and Customize. Personalize. Realize. are trademarks of QUALCOMM Incorporated.

excellent
very good
average
disappointing
don't bother

object that the server will load and use for game-state validation. The interface to the GSAPI proved clumsy; they give you a header file full of C function definitions and tell you to implement them. This is O.K., but it would be preferable to export a structure of function pointers, or implement a class with pure virtual members and export a factory.

In Terazona, sections are called regions, but they operate much like we described the Game State Servers. One interesting thing to note is the fail-over support inherent to their design. Since entities (players or NPCs, generally) are ghosted to the necessary neighbor regions, when a region server fails, another server will reinstantiate the region and attempt to get entity state from servers that had it ghosted. If that fails, it will go back to the database. GSS servers can be brought up and down while the game is running, without restarting other components. Terazona does not have its own implementation of a landscape and does not do landscape validation for you.

Terazona uses the "NPC server as a trusted client" model. The downside of this model is that sometimes the most complicated logic exists in the NPC servers, and their scalability is left up to the developer. The client-server protocol provided is based on TCP, which may be unacceptable for many applications, though Zona has promised a reliable UDP implementation. Because they only ask a small up-front fee in exchange for part of the subscription revenue, Zona's business model is attractive to new MMOG developers. This has the extra benefit of increasing the support and library improvements developers will receive.

Butterfly.net. Butterfly.net is more of a total solution than Terazona, which is not necessarily a good thing. In fact they are proponents of a utility model that allows you to use their server clusters. As traffic goes up and down, those servers are shared with other products that are also using the utility model. Although very cost efficient, this prospect could be frightening to even the most experienced MMOG live teams. They don't prevent you from putting the parts together yourself on your own systems.

The rules management system that the server uses to validate game state uses functions written in Python. This could greatly simplify game authoring and also helps with thread safety. The Python interpreter is built into the game server, so performance shouldn't be a problem.

The Butterfly.net system can have a separate NPC server for every landscape section (which they call locales). If you have an NPC-heavy game, this should help with the distribution of CPU load. (The NPC server is not written in Python.) However, human intervention is required should these NPC servers or locale servers crash. Butterfly.net has no automated server management that can restart processes or bring up another from a pool of servers.

The suggested operating system is Linux with a DB2, Oracle, or Postgres database, but if you were required to use a different operating system or database, Butterfly.net claims that much of the system can be ported. Tools for porting terrain models from Maya or 3DS Max into Butterfly.net's Landscape quad-tree are also provided.

What would we prefer? Although total solutions have their place, MMOGs might be too young of a genre to support this strategy. A more open-ended toolbox approach is proving useful for graphics engines such as Criterion's Renderware, a useful model to emulate on the server-side as well. This would mean a large collection of smaller functional libraries, separated by a well-documented API. Libraries of functions for messaging, persistence, cluster control, and terrain quad trees could be combined into whatever system a developer might need.

In addition, neither system supports the concept of an automated generic pool of servers. This would entail every server being able to perform any function and would create a high amount of reliability with very little human intervention.

Last word. It's very exciting that the massively multiplayer market is growing enough to support the creation of these types of solutions. MMOGs are fast becoming too complex and risky to develop for very much innovation to occur, and we are all happier if systems like Terazona and Butterfly.net help alleviate that. Still, neither has a well-known MMOG that has used their technology yet. Even so, both are strong contenders in this young market.

Visit www.gamasutra.com for a longer version of this review.

TERAZONA

STATS

ZONA
Mountain View, Calif.
(650) 964-1133
www.zona.net

PRICE
Depends on project and studio size.

PROS

1. Portability.
2. Automated game state.
3. Flexible environment model (space, water, ground).

CONS

1. Persistent data is stored in blob format.
2. Reliable UDP still in development.
3. Regions are statically defined.

BUTTERFLY.NET

STATS

BUTTERFLY.NET
Martinsburg, W. V.
(304) 260-9520
www.butterfly.net

PRICE
Depends on project and studio size.

PROS

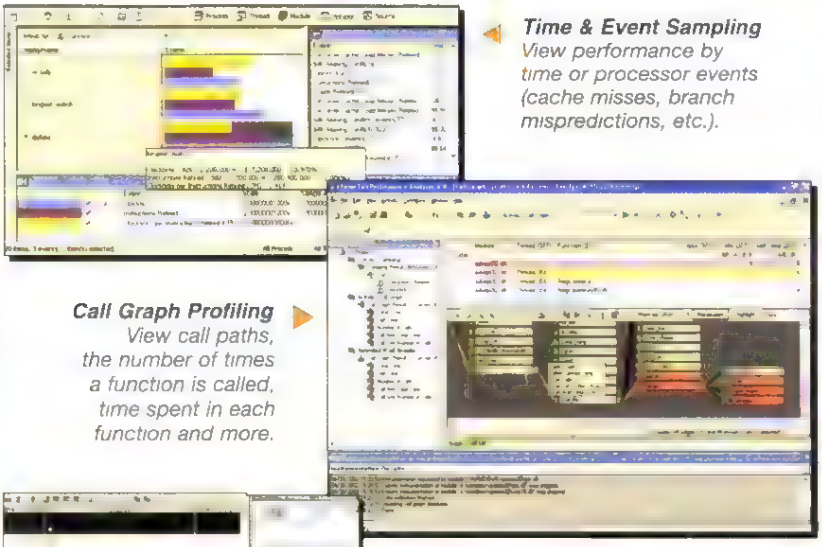
1. Utility model.
2. Almost a complete solution.
3. Persistent data is stored in the clear.

CONS

1. Utility model.
2. Lack of automated disaster recovery.
3. Locales are statically defined.

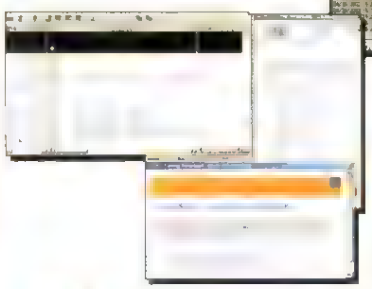
Maximize Your Application Performance with Intel® Software Development Products

Intel, the world's leading processor vendor, provides professional software development tools. These tools are designed to integrate easily with the development environment you're already using. All tools include 1 year of free product upgrades and Intel Premier Support. Take advantage of the advanced features of Intel® Pentium® 4, Intel® Xeon™, and Intel® Itanium® processors with tools from the people who know processors best.



Time & Event Sampling
View performance by time or processor events (cache misses, branch mispredictions, etc.).

Call Graph Profiling
View call paths, the number of times a function is called, time spent in each function and more.



Intel Tuning Assistant
Uses Intel compiler technology to give advice on improving code performance.

Intel® VTune™ Performance Analyzer 6.1

VTune™ Performance Analyzer 6.1 helps you increase your software performance by enabling you to identify and analyze bottlenecks in your code. The VTune analyzer has many powerful analysis features that offer useful guidance on how to improve performance. Version 6.1 supports VS.NET*, C# and multi-threaded applications.

"The Intel VTune Performance Analyzer took a multi-day task and turned it into a sub-day task."

—Randy Camp
V.P. Software Research and Development
MUSICMATCH, Inc.

Intel® C++ and Fortran Compilers for Windows® and Linux®

Intel® has applied its expertise in software optimization for Intel® 32 bit and Intel® Itanium® processor-based systems to develop the next release of its compilers product-line. Version 7 of the Intel® Compilers. The Intel Compilers 7.0 deliver optimal performance on Pentium® 4 and Intel® Xeon™ processors, and the 64-bit Itanium and Itanium 2 processors and take advantage of Multi-processor systems and Hyper Threading technology

Intel® Performance Libraries

The Intel Performance Libraries simplify development of high-performance applications. These libraries free developers from having to write separate versions of code for different processors by providing a simple, consistent API interface that works across processor architectures. The libraries include highly optimized processor-specific code for graphics, multimedia, mathematical and signal processing functions

YOU SAVE UP TO $164	Part No. #	Retail	Discount
Intel® VTune™ Performance Analyzer 6.1	I23 0170	$699	$534.99
Intel® C++ Compiler 7.0 for Windows*	I23 0A10	$399	$305.99
Intel® Math Kernel Library for Windows*	I23 0180	$199	$173.99

* *Also available for Linux*

FREE 30-day trials available at:
programmersparadise.com/intel/gdm

Programmer's Paradise
To order, or request additional information call:
800-423-9990
Email: intel@programmers.com

© 2002 Intel Corporation. Intel, the Intel logo, Pentium, Itanium, Intel Xeon and VTune are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. *Other names and brands may be claimed as the property of others.

03CT02C

Matt Toschlog's Descent into Outrage

Since 1986, Matt Toschlog has enjoyed a varied life within the game development industry. Starting his journey at Sublogic, Matt continued through Looking Glass Technologies, to Parallax Software (which he co-founded), and is now studio director for Outrage Games, a company he splintered off from Parallax and was president of until its recent acquisition by THQ. One of his most notable contributions to videogame history so far includes the DESCENT series.

Outrage's Matt Toschlog

With what looks to be an ongoing trend of big publishers and media conglomerates acquiring once-independent developers, we asked Matt — among other things — to give us an insight on what things are like after the Big Change.

Game Developer. Do you think the current trend of third-party developers being acquired by publishers bodes well or badly for the industry?

Matt Toschlog. It's a good thing in that it allows third-party developers to keep making games. As games, team sizes, and budgets have grown, it's become increasingly hard for independent developers to survive. The stakes are too high, and publishers are hesitant to spend the money required to make a great game if they don't have more control over the process.

GD. How do you see your work changing from the days of Parallax to the present? Has working for THQ changed your work behavior in any way?

MT. The biggest change is that I don't program anymore. When we wrote DESCENT I was one of three principal programmers. Now I'm a full-time manager. Another major change is that our projects are much bigger. At the end of DESCENT we had eight people working on the game; our current project, ALTER ECHO, now has about 25. That means a lot more management and coordination, and a much less intimate environment.

At the time we were acquired, we'd been working on ALTER ECHO for THQ for about a year. When our other publisher ran into trouble, it was natural for THQ to step in. We were in the middle of our E3 crunch when the acquisition happened, and we just kept on working. We had to shift some people around when our other project was cancelled, but for most people, Outrage isn't much different now than before THQ came in.

GD. Do you think the creative process of making a game is affected by these acquisitions? Is there more paperwork, strategy, and playability planning going into the making of a game in a big company than in smaller ones, where things can be a lot more flexible as the game design and programming unfolds?

MT. I think the creative process is affected by the size of the games. If a company is going to spend 5 or 10 or 20 million dollars on a game, they're going to want to keep close tabs on where the money is going. You've got to be a lot more careful when you're spending that kind of money.

GD. What role do you see middleware playing in the future of game development?

MT. Middleware is starting to be a real factor. These days, most consumers don't care as much about technology. Developers realize that and want to focus on content; using stable development systems makes that easier. We've always written our own engines here and will probably continue to do so, but we're placing a high premium on writing reusable code.

GD. You served as chairman for the International Game Developers Association (IGDA) for a couple of years. How do you see the organization's goals compared to when you came on board?

MT. I think the goals of the IGDA have remained pretty constant — to be the voice of the professional developer and to provide support to the people making games. What's changed — steadily, though sometimes slowly — is our ability to fulfill those roles. We've been building membership, establishing programs, making contacts, and generally creating a foundation. Many of us who have been with the IGDA for the last few years got involved because we had a vision of where the association could be down the road. We're still looking toward the future, but we've already come a long way.

GD. Do you miss those garage days when you first started creating DESCENT? Do you ever wish for a return to those simpler days?

MT. Oh, yeah. It was great being hungry and lean like that. Sometimes we talk about doing a GBA game to try and re-create that atmosphere.

GD. Looking back, what elements made the DESCENT series the success that it was? Do you think those elements came together by choice or chance?

MT. DESCENT offered the classic game mechanic of "shoot all the bad guys" in a way that no one had seen before. At the time, 3D games were still somewhat new, and DESCENT was the most 3D of them all. Piloting a ship through those crazy tunnels provided a challenge that rewarded those who could master it.

I think success depends a lot on chance. One thing I've learned is that no matter how hard you work and how well you plan, there's still a huge amount of luck that determines whether your game is fun or if the customers will care. If we really understood how to make hit games, then every game would be a hit.

Cluster F***

Hit the wall yet?
You will.

Unless you build your MMOGs
on the Butterfly Grid,™
you'll be stuck with downtime,
lag and pissed-off gamers.

Only the Butterfly Grid
supports millions of concurrent players and
reacts as quickly as your trigger finger.

Only the Butterfly Grid
slams the cheaters and
drives the most complex AI
at breakneck speeds.

Efficiency. Reliability. Scalability.
Optimal performance.
Break out of the cluster.
Go Grid or go home.

Visit www.butterfly.net
or call 304-260-9520 x9.

Butterfly.net™

Go Grid

Clustering

For the past two months I've been discussing profiling. I started out by decrying the lack of good profiling tools for games; I talked about the difference between a "batch profiler," like VTune or the CodeWarrior profiler, and an "interactive profiler," like the ones we usually build into our games; I then proceeded to make a big wish list of features that a good interactive profiler might provide.

Chief among those features was the ability to abstract the program's timing measurements into higher-level "behaviors" and to do some analysis regarding the frequency and consistency of these behaviors. Modern games don't exhibit one consistent pattern of resource usage. At one time they might be fill-rate limited by the graphics card, at another they'll be slowed down by an abundance of pathfinding and physics. A tool that helps us visualize these patterns would be valuable.

Finding a Clustering Method

Let's look at Table 1, which I introduced in last month's column ("Interactive Profiling, Part 2: Behavior Analysis," January 2003). Here we see three behaviors: behavior A is rendering-heavy, B is physics-heavy, and C is AI-heavy. Imagine the numbers in the table came from a profiling run of a game in progress; as each frame is processed, we get a new set of timing numbers. Now suppose that the next frame is measured, producing the results: `rendering` 72%, `physics_sim` 17%, `ai_pathfind` 11%. Those numbers are very much like behavior A. Even though the numbers do not match exactly, we would expect the profiling tool to classify this frame as exhibiting behavior A.

That's not hard to implement when you have a predefined list of behaviors like Table 1. It's harder when you start from scratch with no preconceived notions of program behavior. In general, we are faced with a clustering problem: given a bunch of vectors containing timing data, we want to arrange the vectors into groups, where the centroid of each group can be used as a summary.

Clustering is a common task in computer science and engineering applications, and many algorithms have been developed, for example so-called *k*-means clustering. Here's how *k*-means works: First, randomly partition your input into a set of initial clusters. Find the centroid of each cluster. These clusters obviously won't be perfect yet, so to fix that problem, you iterate over each input vector and reassign it to the cluster whose centroid is nearest. Now recompute the centroids. Repeat this reassignment process until the results converge, and you perform an entire pass without changing anything.

The name "*k*-means" indicates that there are some number of clusters (*k* of them), and that we are representing each cluster by its center (by the mean of the input values).

Unfortunately, *k*-means and most other clustering methods are batch algorithms. They require you to have all the vectors available at once and to access those vectors randomly. Since I want an interactive profiler, which can classify behaviors on the fly during a single program run, these algorithms are unsuitable.

TABLE 1

Behavior:	A	B	C
`rendering`	75%	25%	25%
`physics_sim`	15%	[illegible]	15%
`ai_pathfind`	10%	10%	60%

Nonbatch Clustering

There's a special term for clustering algorithms that can handle streaming input: they are called "online clustering algorithms" For example, there's an "online *k*-means."

Online *k*-means works like this: Start with *k* cluster centers arbitrarily distributed through space. For each input vector you want to classify, find the closest cluster point, classify the vector into that cluster, and move the cluster point closer to that vector. (See "Radial Basis Function Learning" in For More Information.) Online *k*-means really is an incremental version of batch *k*-means, but making the algorithm incremental is a big change, so the explanations of the two versions sound fairly different.

How do you decide how many clusters are necessary to represent your data stream accurately? There are some heuristics that involve removing cluster points when you have too many (some

JONATHAN BLOW | *Jonathan Blow (jon@number-none.com) really likes the way bedrooms in Korea have heated floors. You just walk around in your socks and it's all warm.*

GameDevelopers Conference Mobile

March 4 – 5, 2003 • San Jose, CA

ONLY AT GDC MOBILE will leaders from the mobile communications industry meet with the world's best game developers. Network operators, content aggregators, technology and infrastructure providers and leading game developers will share information, explore opportunities and discover how to deliver compelling mobile entertainment to millions of consumers worldwide.

GDC MOBILE

www.gdcmobile.com

Keynotes

Takeshi Natsuno
Executive Vice President
NTT DoCoMo i-mode

Ilkka Raiskinen
Senior Vice President
Media & Entertainment, Nokia

GDC Mobile sessions include:

- Beyond Downloadable Games: How New Technologies Will Drive New Forms of Interaction
- Game Design for the Mobile Medium
- Multiplayer Games and Mobile Communities
- Regional Case Study: China Developing Mobile Games in the World's Largest Market
- Regional Case Study: Finland Cross Media Marketing of Mobile Games: Integration with TV, Web and Other Digital Media
- Regional Case Study: USA Development in a Fragmented Landscape
- The Truth Behind Mobile Game Publishing
- Where in the World is LBS? Designing Games Using Geo-Location
- Who Needs A Publisher? Who Needs an Operator?
- World Tour of Mobile Games and Interactive Entertainment

Featuring Mobile Operator Spotlight Presentations from:

Boost Mobile
Cingular Wireless
J-Phone
O2
Orange
Sprint PCS
Telus Mobility
T-Mobile
Verizon Wireless

For more information on GDC Mobile speakers, sessions and events, go to

www.gdcmobile.com

Sponsors:

NOKIA

MOTOROLA intelligence everywhere

GamaNetwork

of them are lying idle, never attracting input) and adding cluster points when you have too few (when the radius of space attributed to a particular cluster point is too large). Unfortunately, these heuristics all require some kind of a prior knowledge of the data: you need to know how many samples is the minimum for considering a point to represent a valid cluster, or how spatially large a good cluster is likely to be. In other words, you need a good guess about the duration and scale of your data.

But we don't want to have to guess at that stuff in order to get a good profile. Really, we want the computer to figure out appropriate values from context. And those values may need to change. If we are running our game for five minutes and see some frames that spend 70 percent rendering, 75 percent rendering, or 80 percent rendering, we probably want to draw distinctions between these and call them three different behaviors. But then if we walk into another room in the game and we start seeing a lot of behaviors like B and C from Table 1, then perhaps all of the previous frames should be categorized as A.

Clearly, some kind of hierarchical clustering is necessary. It's probably best to let the user decide at which detail level to view the abstractions; we should maintain a tree of behaviors from which the user selects a particular view. Unfortunately, the typical methods, such as hierarchical agglomerative clustering, tend to be data-intensive batch processes.

Visualization

There's one other problem we need to solve for effective visualization of our profiling data. In a full-sized game, there may be a few hundred different subareas of the program for which we get profiling data. In clustering terms, we're clustering data points that have a few hundred dimensions each. Suppose we manage to do this, and we get a result with 15 different behavior clusters. Those clusters are still points in a high-dimensional space. We can list them, like in Table 1, but we'd like better methods of visualization. Maybe we want to project these clusters somehow onto a 2D chart and have the clusters that are most alike placed near each other on that chart. Maybe the two dimensions of the chart can even mean something, related to the two biggest ways in which the behaviors tend to vary.

There are some well-known methods for projecting multidimensional data onto a 2-plane. These methods tend to fall into two categories: ad hoc, producing poor results; and complicated, slow, and scary. There's another alternative, though: a hybrid kind of algorithm that clusters and projects simultaneously, and does a good job of both.

The Self-Organizing Map

In this month's sample code, I used this hybrid algorithm, the Kohonen self-organizing map (SOM) to perform the classification of CPU behavior. The SOM is an array of points in the input space; all the points are initialized to random values. For each incoming frame of profiling data, we find the closest point in the SOM. We then move that closest-match point toward the position represented by the incoming frame, just like online *k*-means. But the SOM does something extra: in addition to giving the clusters positions in the input space, it treats them as having positions in another space also. The clusters have fixed positions along a regular grid, in what I will call "neighbor space." Whenever you move a cluster center toward an input in input space, you also look up its neighbors in neighbor space and move those neighbors through input space in the same direction, thought not by as much.

This neighbor space is usually implemented just by storing the cluster data in a 2D array and by iterating across nearby cells in that array. You can think of this as applying a "move filter" to the array, where the filter kernel is high in the center but tapers off with an increasing radius.

After many such iterations, the clusters in input space will converge, just like *k*-means. The size of the move filter shrinks over time as the clusters become settled, until it becomes a point. The "self-organizing" part of SOM represents the idea that, because we applied that move filter in neighborhood space, we were constantly influencing neighboring array cells to take on similar values. So after we are done processing our inputs, we not only get cluster centers, but the clusters are arranged in the 2D grid according to similarity. We can generate a 2D display where similar clusters are drawn near each other, which can help us understand the data.

This is a quick-and-dirty description of the SOM, with some simplifications. For reasons of isotropy, like we saw when looking at networking ("Transmitting Vectors," July 2002), it's usually better to use a hexagonal tiling than a rectangular grid. Also, the SOM grid can consist of any number of dimensions, but two dimensions is the most common choice, since a map with more dimensions is harder to visualize. Finally, initialization of the SOM is best done using some attributes of the training data, since a random spread in a high-dimensional space is likely to produce points that are not near anything in your data set. In this case, the only way array elements are used is when they get dragged into use by a neighbor, so it takes a while before the SOM is working at reasonable effectiveness.

For some references that explain self-organizing maps in greater detail, see For More Information.

Adapting the SOM to Real-Time Tasks

The SOM is a batch algorithm for two reasons. For one, to produce good results, the algorithm wants a randomly ordered sampling of the data. Imagine first presenting all your samples of behavior A, then presenting an equal number of samples of behavior B, which happens to land in a cluster neighboring A, so the movement filter for B wipes out most of A — this process will not produce desirable results.

The second reason the SOM is a batch algorithm is that it wants to have a global notion of time. Early in training, the

GamaNetwork

MAKE BETTER GAMES

JOIN THE WORLD'S LEADING GAME DEVELOPERS TO SET THE STANDARD FOR THE NEXT GENERATION OF GAMES. FIVE DAYS OF INTENSIVE SESSIONS PROVIDE INFORMATION AND INSPIRATION IN VISUAL ARTS, GAME DESIGN, PROGRAMMING, AUDIO, BUSINESS & LEGAL, AND PRODUCTION.

GameDevelopers Conference

March 4-8, 2003 • San Jose, CA

movement filter is very large; over time it shrinks, until toward the end of training it's almost a point. But with live input data, we don't know how long we will be running, and we may wish to run indefinitely. So I made some modifications to the SOM algorithm to adapt it for real-time data. These modifications are somewhat questionable, since I have not analyzed them deeply, and I can already think of some ways they may be problematic. But to a first approximation they work O.K.

Shrinking the Filter

Rather than using a global time, I introduced the idea of "local convergence," which is a value between 0 and 1. The filter size centered around any given array cell depends on that cell's local convergence: a convergence of 0 means a big filter, 1 means a small filter.

Whenever a point in the SOM is generally staying in the same place, its convergence value increases. If it starts getting yanked someplace else, its convergence value decreases. I use the IIR filters I described in "Toward Better Scripting, Part 1" (October 2002) to track the trend of a point's motion. If many training steps confirm the positions of some region of the map, that region will no longer feel the need to upset its neighbors. But if an area of the map keeps being upset, its "local time" will go backward and it will jostle its neighbors in a bid to create more space for the competing values.

This "hardening of the map" does to some extent alleviate the "B wipes out A" problem but does not solve it. To solve it, I would implement a hierarchical series of maps, with maps that change quickly, feeding data into maps that change more slowly. I haven't had time to try this yet, but I plan to soon.

One further problem is that there's no way for the SOM to grow. To achieve reasonable results, you need to construct it with enough initial nodes. But I believe hierarchical maps would solve this problem as well.

Initialization

I wanted a good way to initialize the SOM. As I mentioned earlier, random initialization is not so good. The way it's usually done for batch data is to perform principal component analysis on all the input, find the two non-axis-aligned dimensions along which it has the greatest spread, and initialize the SOM array entries to cover that spread evenly. In other words, you find the covariance of the data with respect to the mean, then pick the two longest axes of the multidimensional ellipsoid. This process is just an *n*-dimensional version of what I demonstrated in "My Friend, The Covariance Body" (September 2002).

Since I don't want to wait for the whole batch of input data, I collect 10 frames' worth of "warm up points" before the SOM kicks into training mode. Then, because I didn't want to write *n*-dimensional matrix decomposition code, I approximate the spread as follows: First I pick a random warm-up point, then I find the warm-up point that is furthest from it. I find the vector between them, and that is the longest axis of the spread. Then I remove that dimension from the warm-up data by subtracting each point's projection along that axis. Then I choose another random point, find the warm up point furthest from that one, and that is the second-longest axis of the spread. I then initialize the SOM entries to points evenly spaced across the plane defined by these two axes, centered on the mean. If the axes are degenerate within some numerical threshold, then I seed the SOM with random offsets from the mean.

Sample Code

This month's sample code (available at www.gdmag.com) is an updated version of the toy game world that I described in my December column ("Interactive Profiling, Part 1"). In addition to AI-heavy and entity-render-heavy behavior modes, there's now a sun in the sky. When you look toward the sun, ridiculous amounts of lens flare are drawn, bogging the system down.

As you move around the game world, you'll cause various behaviors that are mixtures of AI, entity, and lens flare. Whenever `mouselook` is enabled so that you can move around, the game system takes each frame's profiling data and classifies it using an SOM. When `mouselook` is disabled and the SOM is paused, you can move the mouse pointer over the SOM's display and look at the behaviors it has detected.

There are a number of visualizations enabled, which are explained in the README. These visualizations aren't as intuitive as I'd like, but after a bit of exposure you get used to them and can infer interesting relationships about the clusters by looking at colored diagrams.

Future Work

So far this work has been pretty experimental, but I think it shows some promise. I am going to try some hierarchical versions of the SOM and online *k*-means and will report on their effectiveness in a future column. In the meantime, next month I'll go back to a more conventional topic: creating better-looking graphics.

FOR MORE INFORMATION

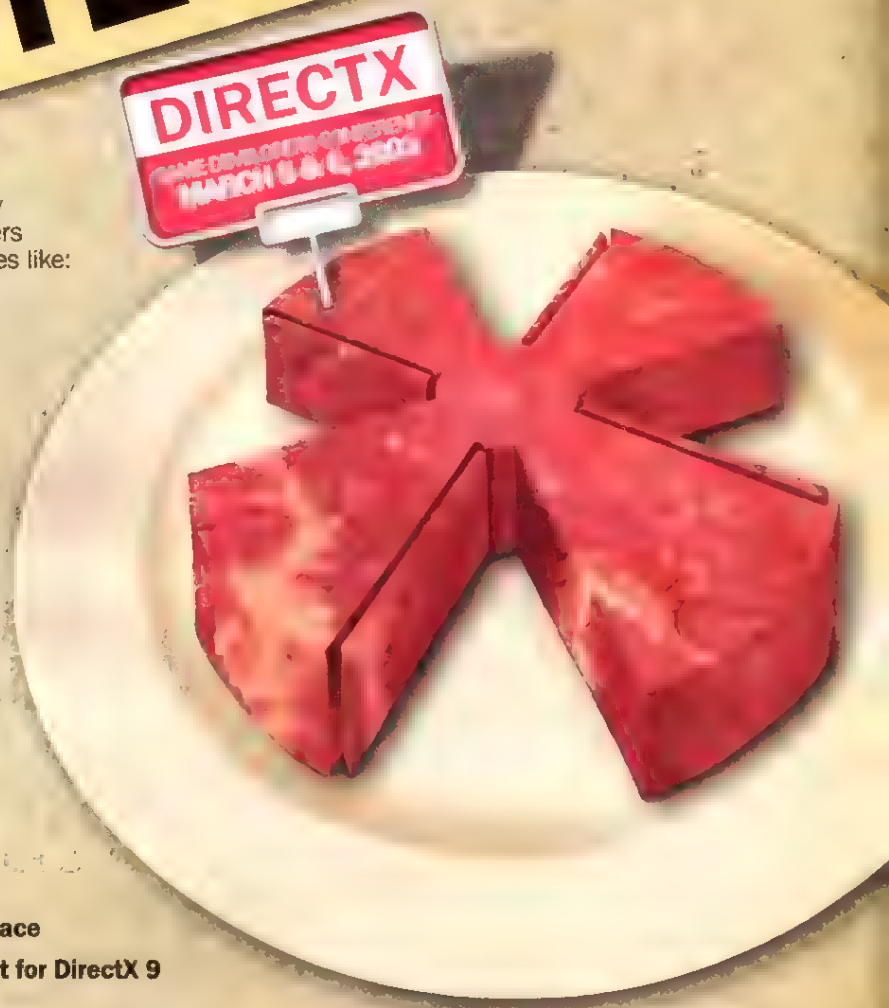
Kohonen, Teuvo. "The Self-Organizing Map (SOM)"
www.cis.hut.fi/projects/somtoolbox/theory/somalgorithm.shtml

Moore, Andrew W. "K-Means and Hierarchical Clustering"
www.cs.cmu.edu/~awm/tutorials/kmeans09.pdf

Palmeri, Paolo. "Radial Basis Function Learning"
http://mpes.cnuce.cnr.it/~palmeri/datam/articles/okmcps.gz

Vesanto, Juha, Johan Kimberg, and others. "Enhancing SOM based Data Visualization" www.cis.hut.fi/projects/ide/publications.html#vzjka98

NOTHING BUT MEAT!



MICROSOFT DIRECTX DEVELOPER DAY WEDNESDAY, MARCH 5, 2003

Register for Microsoft® DirectX® Developer Day Sponsored Tutorial during the Game Developers Conference and learn about prime technologies like:

- **High-Level Shader Language**
- **Nontraditional Lighting and Techniques**
- **Ultimate Rendering Realism**
  - **Spherical Harmonics**
  - **High Dynamic Range Techniques**
  - **Latest Innovations from Graphics Research**
- **Tools and Technologies to Make Writing Games for Windows® Easier and Better**
- **Power Debugging with Microsoft Visual C++® and the DirectX Extensions**
- **Interactive Analysis of Windows-based Games with D3DSpy**

MICROSOFT SPONSORED SESSIONS THURSDAY, MARCH 6, 2003

Get a taste of these beefy Sponsored Sessions that pack a lot of meat:

- **Opportunities in the PC Gaming Marketplace**
- **Managed DirectX: Microsoft .NET support for DirectX 9**
- **Screaming at the Machine: Using Speech Recognition as a Complement to Traditional Game Input Techniques**
- **Best Practices for DirectX 9 Audio Programming**
- **How to Write a Network Layer**

Sponsored By Microsoft | best When Used On 3/5/03 & 3/6/03

Register online for the Microsoft DirectX Developer Day and the Sponsored Sessions at:

www.gdconf.com

Get the latest on DirectX at **http://msdn.microsoft.com/directx/**

Microsoft® Windows® Gaming

© 2003 Microsoft Corporation. All rights reserved.
Microsoft, DirectX, Visual C++, Windows, and the Windows logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Microsoft®

What's Wrong with Our Games? Part 2

Illustration by Dominic Bugatto

I have heard many times, expressed in a variety of ways, that the success of a game lies firmly with its execution rather than in its design. It is a fact that many ideas and their initial design documents put forward what seems to be an interesting and exciting concept, but by the time the game finds itself on the shelf, it has turned into nothing more than a mediocre space-filler that will find a home in the bargain bins within the month.

I'm sure that any experienced developer reading my column could write me a list as long as my arm of the reasons why a finished game differs substantially from the original design spec and why, in general terms, evolution of ideas through the development process is always a good thing. Implementation, rather than initial ideas, usually accounts for the eventual quality of a game when it is released.

There is no area more relevant to the discussion of implementation than that of a game's visuals. The irrelevant language of a pitch document that promises a game "set in richly detailed fantasy world of epic proportions" has to translate meaningfully onto the screen to have any chance of being even close to the truth. If you run such a document through a truth filter, many such sentences would read, "Set in a routinely realized fantasy cliché, made to appear large by repeating the same geometry over 100 times."

Continuing the theme I began in last month's column ("What's Wrong with Our Games? Part 1," January 2003), in which I covered design, animation, and character creation, this month I'll look at some additional areas in game visuals that often lose something in the translation from page to screen.

Lighting

One indication of how videogame technology is improving is the quality of the lighting available to designers. Unfortunately, many games still view lighting as number 138 on their list of things to pay close attention to, and as a result, the energy developers spend creating a world of detail and quality is wasted for many games. They pay little attention to lighting, or decide not to invest much time in their light-related technology. Nasty lighting can hammer even a well-constructed scene to death, whereas high-quality lighting will bring a scene to life. Attempting to re-create the real-world behavior of light in a 3D scene has led to a whole variety of

HAYDEN DUVALL | *Hayden started work in 1987, creating airbrushed artwork for the games industry. Over the next eight years, Hayden continued as a freelance artist and lectured in psychology at Perth College in Scotland. Hayden now lives in Bristol, England, with his wife, Leah, and their four children, where he is lead artist at Confounding Factor.*

Are You Sweating Yet?

You don't have to sweat bullets trying to figure out ways to reduce the risk of development and operation overhead for Massive Multiplayer Online Games. Let our team of MMOG development and infrastructure experts help you solve the complex challenges – while you focus on the GAME. Now you can deliver awesome player experiences while spending a lot less time and money on network infrastructure issues.

THE DESTINY OF YOUR MMOG PROJECT IS IN YOUR HANDS...

Zona,Inc.
www.terazona.com
Phone: 650.964.1133
E-mail to: sales@zona.net

TERAZONA

lighting solutions over the years.

Initially, severe color-palette restrictions made the idea of lighting irrelevant, but as we moved through the 8-bit and then the 16 bit era, the simulation of lighting effects allowed our early 3D worlds to appear more solid. Once true 3D began to appear, it wasn't long before basic light sourcing was a common feature. Today, artists can apply just about any high-end lighting solution to some degree in a gaming context, and the question of which solution to use then depends on cost versus benefit and which platform you're dealing with.

Up until recently, the most advanced of these solutions have been exceptionally slow. Unless you had access to a render farm the size of Alaska, the massive amount of numbers being crunched manifested themselves in single frame render times that were measured in weeks rather than minutes.

Artists now have the tools and the technology to do more than place colored point lights around a scene that they hope will do the job. The approximation of real light distribution is now more accessible within the limitations of a game world. Designers can now use texture baking to incorporate the subtleties of radiosity, global illumination, and lighting from high-dynamic-range images.

Texture baking has been around for a while, but the newest iteration of many high-end renderers now brings together the most advanced lighting solutions with the ability to render this light information to textures. Whether the light information is rendered into a separate layer to be added on top of more traditional texturing or added directly to the textures themselves, the results one can achieve are far more realistic.

The downside is that texture baking can destroy texture memory in no time flat. The price to pay for what is effectively light distribution information, customized for each surface, is that the cost to memory can be massive in scenes with a huge amount of surfaces. Condensing multiple light maps into more economic single textures, applying low-resolution light maps on top of regular texturing, and careful level design that takes these limitations into account can all make texture baking more usable. While the cost is high, the quality of the end result can be worth it.

Heavy-handed application of just about anything, regardless of how nice it is, can ultimately do more harm than good.

Textures

The easiest targets for game artists to pick on when examining the work of their peers are most probably the textures and the texturing. Many games now have extremely large worlds that can be explored in fairly close detail, and as a result, it is hard to find a game that has faultless texture work across its entirety.

That said, there are a few texture problems that appear time and again in games:

Bad mapping. Bad mapping can take many forms, some worse than others, and usually indicates either a lack of time and resources, or some level of apathy on the part of the offending artist.

Faced with a significantly large amount of geometry to texture in a given time, it is understandable that inconsistencies should appear. Texture streaking from applying a planar map across faces that are nearly perpendicular to the UV plane is one example of careless or hurried mapping that looks very ugly.

Seams that show painfully obvious joins between mapped sections can also stand out, especially if the geometry is large, like a rock face for example. With many complex objects, especially organic shapes, it's all but impossible to map the entire surface with a continuous, seam-free texture, so some jiggery-pokery is often needed to reduce the appearance of problems.

Many designers overlook the option of placing joins in the texturing where they will be the least visible. This may seem like an obvious solution, but it's important to remember that "the least visible" means from the player's perspective, not that of the artist, so developers must play through the sections in question carefully to assess the least visible areas for these scenes.

Another crude but nonetheless worthwhile problem-masking technique is using textures that naturally minimize the appearance of seams. Such textures are those that are even in terms of detail, color, and contrast. Ideally, a system that obscures unavoidable seams, such as multitexturing, works better, as long as the artist has the time and the inclination to take advantage of this method.

Specular and bump mapping. Specular and bump mapping technologies are not new, but they are only now becoming practical in any real sense for mainstream gaming. Still, heavy-handed application of just about anything, regardless of how nice it is, can ultimately do more harm than good. Both bump mapping and specular effects have the ability to introduce additional and dynamically responsive material properties to surfaces within your game. But if they are applied with no concept of restraint, they will end up being intrusive gimmicks that turn a game's visuals into an unsuccessful mess.

Coupled with dynamic light and shadowing, bump mapping is a particularly useful way to achieve the appearance of

fine surface details without using actual geometry. But many designers are drawn into making just about everything bumpy, or at least into adding bumps where they have no real need to be. Likewise, specular maps can seduce artists into finding ways to incorporate specular effects on every object they create, not a good idea in most circumstances.

Bottom line: Restraint is key, as the player's attention is drawn to the areas where the techniques I've just mentioned are used judiciously, rather than to excess.

Straight from scan. Any game that is attempting a certain level of realism is likely to use photographic sources in its texture generation. Problems arise when the initial scan or digital photo finds its way into the game without any significant processing. The end result of unprocessed images is usually textures that feel out of place or flat. As most textures need to be balanced for brightness, contrast, and other properties to make them consistent with the game world, neglecting to process images will usually make them stand out unnecessarily.

Scaling gone wrong. Another common problem arises when textures are completely out of proportion with the environment that they are supposed to be part of. This problem is especially common in larger areas, where a texture is likely to be repeated many times, and artists sometimes cut down the number of repeats by scaling a texture up.

The problem becomes most apparent when a character (or player, if it's third-person) is in close proximity to the offending texture, where it is possible to see that the heads of the nails in the wooden planks are actually the size of watermelons. This kind of scaling mismatch is easy enough to avoid, but it requires the artist concerned to pay close attention once again to the player's experience, as opposed to building and texturing geometry in isolation.

Environment mapping. The use of an environment map, particularly in conjunction with other layers of texturing in order to give the appearance that a surface is reflective, has become one of the most abused effects of recent times. It is a toy that needs to be put back into its box and only brought out when it genuinely fits.

Especially bad offenders combine environment maps with such effects as worn and damaged metal surfaces, so that a material that should be relatively unreflective ends up looking like it has been coated with a very thick layer of varnish and buffed to a mirrorlike shine. Here again, restraint is key.

Cutscenes

The art of creating the perfect cutscene is a large enough topic to warrant several columns, but when asking others what they think most often goes wrong with cutscenes in a game, the almost universal answer is that they are too long.

The debate about what the role of the cutscene actually should be in a game has several contrasting viewpoints. Still, no matter how cutscenes are presented, whether in-engine, 100 percent prerendered, partially interactive, or whatever, the bottom line should always be: do they improve the player's experience, or do they just get in the way of enjoying the game?

This can be a difficult question to answer, as my idea of a great cutscene could be the opposite of what you are looking for. However, because lots of people feel that long cutscenes intrude on the experience of playing a game, it would be a good idea to look at ways of reducing cutscene length where possible without compromising story or atmosphere. There is a limit to how much information a player can retain from a cutscene, so if it does contain things they need to remember, then dragging it on will make that more difficult.

Geometry

Now that polygon counts are less of an issue than they were a few years ago, such problems as having to use five-sided cylinders or being forced to resort to triangular cross-sections everywhere don't exert the pressure they once did. Building environments that look good isn't suddenly a piece of cake just because we can be a little freer with the triangles, but the problems at present tend to focus more on such elements as texturing, lighting, or general design issues.

In this respect, it isn't the quality of the geometry that is a problem; it often has more to do with the variety of geometry used, particularly when it comes to interior spaces. It's very easy to decide that the corridors of a space station would, in reality, all be pretty much identical, so why not just take a section of geometry that you are happy with and duplicate it as many times as necessary? Likewise, the sewers under the castle are all made from the same brick, so why build 10 variations of the same thing?

Repetition is a great time saver, and when you use it with care it can be virtually invisible to the player. On the other hand, without enough unique geometry or some other form of land marking, a player can get lost quite easily. Navigating an area may seem simple to the artist who created it, but often the modeler's perspective is completely different from the player's restricted viewpoint. For the player, figuring out how an area is laid out, even somewhere that is relatively uncomplicated, may prove to be quite a task.

Aside from the possibility of becoming disoriented, a player can also bore quickly if the visuals become overly monotonous. The artist has failed in his or her task if players switch off because they are tired of wandering around locations that all seem to be essentially the same thing.

In the end, we can't expect our games to be 100 percent perfect in every respect. But in the quest for greatness, the most successful approach will be to spend as much time as is available to us to improve the areas that will make the most difference to the player's enjoyment of a game.

Music That Won't Give You a Rash: Taking the Irritation out of Repetition

No matter how much you love your favorite piece of music, if you listen to it enough times, the repetition becomes annoying. In-game music is subject to this kind of repetition, so how do you avoid irritating listeners? Given that repeating elements is a fundamental part of composition, let's look at how the game's audio engine affects the fundamentals.

Repetition in the structure. Music is typically structured in patterns. For example:

A-B-A-B-C-B

denotes a typical rock-song structure, where A is the verse, B is the chorus, and C is the bridge. Notice that B actually takes up half the piece — this is done purposefully so that by the end of the piece the listener should be able to hum the chorus.

Even though the point is to hammer home the chorus, there's a reason the song doesn't simply feature the chorus all the way through. The contrast of other musical sections is required to avoid boring the listener. Although there are examples of 20th century music that push the envelope in the realms of contrast and repetition, a good balance is required for the most part.

But what happens to this balance when a game's audio engine loops a piece of music? Our example structure would end up looking like this:

A-B-A-B-C-B — A-B-A-B-C-B — A-B-A-B-C-B — and so on.

In just three loops we've heard B nine times and we're already sick of it. However, we can remedy this situation by breaking some of the traditional rules of composition.

The trick is to write with the structural repetition reduced to the point where the piece needs to be heard many times in order to "understand" it. An extreme example of such a structure might be:

A B C D E

A music professor would give such a piece an F, because in theory it would lack coherency. However, when the game engine repeats the piece over and over while you're killing monsters, you eventually become familiar with all five parts. It would take considerably more repetitions than the rock-song example to get to know this piece, which is exactly what we want.

For most composers, it's quite hard to write something completely lacking in repetition. They have a deeply ingrained instinct to use repetition, so it's better to try easing into it rather than giving it up cold turkey. If we took the rock-song structure, wrote an additional D section, and varied one of the A sections to give:

A-B-A′-D-C-B

we'd have a tune that is much more loop-friendly without being completely incoherent.

Repetition in the melody. Just as the use of structure needs to be reconsidered in the context of in game repetition, so does that of melody. A melody or theme is the most commonly remembered element of a piece of music. While the arrangement and groove may be pleasing, the melody is what the listener actually remembers. In a game, this can be bad — every time that theme pops up, the player notices it.

Themes are built using a structure. John Williams' opening theme for *Star Wars* uses a single rhythmic motif repeated four times with a variation on the last repetition. It's brilliantly crafted and highly memorable — too memorable, in fact, to be heard many times in succession.

Many of the preceding suggestions regarding the structure of a piece of music can be applied to creating themes. For example, you could try increasing the variety in the motifs that make up the theme. However, we'd like to make the case for not using full themes at all in games. Rather than making a coherent musical sentence using short motifs, try leaving the pieces unglued, peppering your piece with smaller fragments instead. By varying these motifs so that you seldom hear the same one twice, you can achieve musical coherency without hitting the listener over the head repeatedly with a single, attention-grabbing melody.

The final mix. The result of all this reduced repetition will be music that plays well over the course of playing a game but doesn't stand up very well to a single listen. If you're presenting music tracks independently of the game, such as on a CD, you may need to create different versions of the pieces that incorporate a more usual level of repetition. I never promised that better game music was going to be easy.

SIMON AMARASINGHAM | *Simon is a composer/sound designer with dSonic (www.dsonicaudio.com), a company that creates music and audio specifically for the game industry. Kemal Amarasingham and Vincent Stefanelli, who are also founding partners of dSonic, were contributors and consultants for this article. dSonic's past credits include* System Shock 2, Arx Fatalis, *and* Masters of Orion 3.

Game Development. It's Serious Business.™

"Game programming is without a doubt the most intellectually challenging field of Computer Science in the world. It's the fusion of art, science, mathematics, fiction, and more. A game programming book should be an exciting adventure for both the author and the reader."

—André LaMothe, Series Editor

The Premier Press Game Development Series:™

- Game Design: The Art & Business of Creating Games
- 2D Artwork and 3D Modeling for Game Artists
- Beginning Direct3D® Game Programming
- Data Structures for Game Programmers
- Game Scripting Mastery
- OpenGL Game Programming
- Swords & Circuitry: A Designer's Guide to Computer Role-Playing Games
- The Zen of Direct3D® Game Programming
- Multiplayer Game Programming

Premier Press
A Division of Course Technology
www.premierpressbooks.com

Available at your favorite bookstore or call 1.800.842.3636 to order.

Good Points Don't Go Down

One of my favorite jobs as a freelance designer/producer is doing game/studio reviews for publishers. Working on a design on my own can be an isolating experience, so I relish the opportunity to get out in the world to visit a development studio and essentially do a tune-up, advising them on business plan, company or game team structure, and tools and techniques to manage their games to completion. But my favorite part of that experience is tinkering with the design, looking for ways to help them improve the game while minimizing production costs and risks.

I've found the growing body of game rules very helpful in this process. I often switch roles from sage to scholar as someone makes a suggestion about the game design, and I think, "Rule!" and jot it down. One such moment came during a meeting at Firetoad Software in Calgary, reminding me of one of the very first rules I learned in the arcade business.

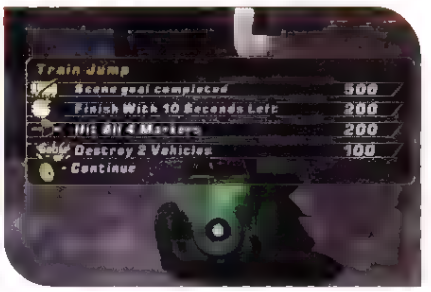
The Rule: Don't take away points or other hard-won possessions from the player.

It's often tempting to design a penalty for the player to emphasize failure at a task or to discourage the player from attempting to do something in the game you don't like. But failing and being discouraged just aren't fun. There's always a way to turn it around and reward the player for success, or encourage them to do what you want.

The Rule's domain. This rule applies to all games but is particularly important where points are awarded, as in sports games.

Rules that it trumps. This rule should trump the temptation to take things away from the player for reasons of realism or to add excitement.

Rules that it is trumped by. Make villains and opponents evil to the player, not just the character. There are cases where it can be a good thing to take things away from the player — a villain in a game who is out to destroy the world is well and good, but when he takes away your character's best magic sword, he's really evil! The caveat is that in defeating (or at least challenging) that villain it becomes possible to win back whatever you have lost. This technique has been used effectively in games as far back as the thief in ZORK, or the temporary loss of coins or rings in many platform games, or as recently as Daxter's transformation in JAK & DAXTER.



The game CHASE awards points for success rather than taking them away for failure.

Examples and counterexamples. Let's consider two frequent violations of this rule. First, we have racing games that use timers to deduct from the score when the player is slow to finish, sometimes to the extent that it is impossible — dare I say pointless? — to try to finish the race. Why not, as in the Xbox game CHASE from I Imagine, give a bonus when the player finishes early? Or use a countdown bonus timer that counts more slowly with time, so there's always some bonus, but one that increases geometrically the faster the player finishes. Second, in some shooting games when the player hits a team member or noncombatant, they turn on him in anger, and points are deducted. It's just as easy, and more fun for the player, to award a bonus for *not* hitting them, or even award the player with extra help or gifts as appropriate from gratefully spared civilians.

Tidbits from the e-mailbox. Dave Grossman was one of many who responded to the "Godfather Paradox" (November 2002) column. He offers some rules that I'll paraphrase for brevity:

- Have a realistic budget: Don't assume it can be done for less than the original.
- Cut some good stuff: Make room for new innovation by resisting the temptation to rehash all the good bits of the original.
- Pretend it's not a sequel: Challenge yourself to think of it as an original game.

Like a teenager moving out of the house, says Dave, the sequel must find its own path. His credentials include DAY OF THE TENTACLE, an excellent example of his rules.

Finally, Ray Mazza, a graduate student at Carnegie Mellon University, suggests using popular musical themes from original games and building on them in the sequels, and similarly using some recognizable territory as a part of the new, larger world. That reminded me of how nicely DIABLO II did both of those things by including the town of Tristam along with its signature musical theme.

NOAH FALSTEIN | *Noah is a 22-year veteran of the game industry. You can find a list of his credits and other information at [illegible]. If you're an experienced game designer interested in contributing to The 400 Project, please e-mail Noah at noah@theinspiracy.com (include your game design background) for more information about how to submit rules.*

Did a bloated .ELF crash your holiday party?

"CodeWarrior's short compile and link times during development have improved our ability to stay on schedule and ship on time!"
Craig Broadbooks, Solworks

Start the New Year off right! CodeWarrior™ development tools from Metrowerks shrink your bloated .ELF and won't weigh down game development.

Did a big, fat .ELF block your games from getting out the door last year? Make a resolution to get your execution files back down to a normal size, so you can ship on time next year. You can do it with the CodeWarrior™ compiler — without forcing yourself to trash that sweet game code. Save even more time with CodeWarrior Analysis Tools that pinpoint and solve additional performance bottlenecks. And this year, lose the bloated .ELF!

CodeWarrior™
Game Development Tools

Try an evaluation copy and ship on time next Christmas!
Request an eval at www.metrowerks.com/games or call +1 512-997-4700, within the U.S. call 1-800-377-5416

NINTENDO GAMECUBE®
and Game Boy® Advance
tools and middleware vendor

Tools & Middleware

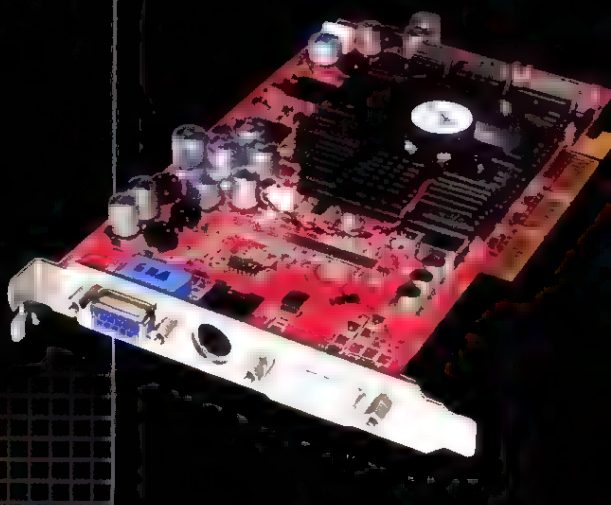
© 2003. Metrowerks, Corp., a Motorola Company. All rights reserved. Metrowerks, the Metrowerks logo and CodeWarrior are registered trademarks of Metrowerks, Corp. in the U.S. and other countries.
NINTENDO GAMECUBE and Game Boy are registered trademarks of Nintendo Co., Ltd. PlayStation is a registered trademark of Sony Inc. All other trademarks remain the property of their respective owners. MSG01401A

metrowerks
Software Starts Here

JOHN LALLY | *John is the animation technical director at Insomniac Games. In addition to* RATCHET & CLANK, *John has animated characters for* SPYRO *2 and 3 and for several Squaresoft Games. He is currently working on Insomniac's newest Playstation 2 project and can be contacted at jpl@insomniacgames.com.*

Radeon 9700

Heralding The Next Generation Of Programmable VPUs



The new chip is significantly faster than anything available in the market. But ATI is keen to stress that this is only one of its key new features. The real gain is not just in the increased number of pixels thrown at the screen, but as ATI describes it, the 'quality per pixel' visible on the screen. Leading PC games developers like Id's John Carmack and Epic's Tim Sweeney have been demanding this emphasis on quality for sometime.

INTRODUCTION

Back dur ng GDC 2001, the seem ngly age o d question of PCs versus consoles cropped up again, but th s time w th more apocalypt c overtones for the fans of the big beige box The seminar featured a roundtable with the doom-laden title "Is the PC really dead? Moderated by 3DO's Trip Hawk ns, the sess on featured a motley co lection of industry big names like Microsoft's Ed Fries and Sony's ever-eloquent Phil Harr son

With the new consoles boasting 3D graphics equa to or better than a h gh-end PC, some of the speakers ventured to pred ct a steady dec ne for the PC as a gam ng platform The more wizened attendees disagreed, po nt ng out the difference n gamep ay att tudes between PC and console gamers.

At the same time, 3D graph cs ch p manufacturers were assuring us that t was only a matter of time before the PC regained ts crown Even Nv d a who makes the graphics ch p for the X-Box, the most powerfu conso e n the market, admitted that the PC w soon be lead ng the f ock aga n

With the aunch of the Radeon 9700, this pred ction has now come true F tt ng y enough this conso e bash ng 3D graph cs ch p has been ntroduced by Nvidia's arch-rival, AT Technologies Inc

Like the x86 CPU market the 3D graph cs ch p industry s fast consol dat ng into a two-horse race For a few years now, on y AT and Nv d a have been able to cons stently update their product ineup every six to nine months This bruta schedu e has seen the l kes of 3dfx, S3 and Rend t on fa l from glory and get acquired by bigger p ayers Others ke Matrox and 3Dlabs have seen their market share significantly reduced

The new ch p s signif cant y faster than anyth ng ava lab e in the market But AT s keen to stress that this is only one of ts key new features *The real gain is not just in the increased number of pixels thrown at the screen, but as ATI describes it, the 'quality per pixel' visible on the screen* Lead ng PC games deve opers ke Id's John Carmack and Epic s T m Sweeney have been demanding this emphasis on qua ty for sometime

n other words, the message to the PC hardware cognoscent is that the future of PC gaming s about mage qua ty and performance rather than just the atter And for the time be ng at least the Radeon 9700 has aga n made ATI the unquestioned eader n both performance and mage qual ty

RADEON 9700 FEATURES EXPLAINED

Performance

Even n th s age of g gatexel graphics ch ps performance matters Wh e the graphics hardware connoisseur might sa vate at the prospect of increased frames per second n the r favourite first person shooter, PC games deve opers perce ve an opportunity for better shader effects n their efforts to create rea istic and even non-rea ist c gam ng rea ms

The Radeon 9700 s the f rst graph cs chip to have e ght rendering p pel nes Each rendering pipe ne is equipped w th one textur ng un t, g v ng the new ch p the ab ty to process up to 8 p xels n a s ng e clock cycle W th a core c ock speed of 325MHz the ATI chip has a max mum theoretical f l rate of 2 6 b l on pixels per second. Th s s more than doub e the fil rate of ts predecessor, the Radeon 8500

Al this mpressive horsepower would be mean ng ess if the graph cs ch p tself s unable to retr eve data from the frame buffer as qu ck y as required

To fac l tate th s, *the Radeon 9700 has been given a 256 bit memory interface which provides a maximum bandwidth of more than 20Gb/sec This is more than double the effective bandw dth available to the previous generat on of graphics chips.* The ch p supports up to 256Mb of DDR memory

As bef tting a next generation graph cs ch p, polygon throughput has been sign f cantly bumped up as we ATI c aims a theoretica maximum throughput of around 300 m l on fu y transformed and t po ygons per second But remember this is a theoret ca number, w th rea world performance l kely to be below th s

For nstance, sustained po ygon throughput is usua y affected by the number of vertex shader effects that a programmer dec des to use n a particular scene t cou d also be mpacted if data needs to be fetched from system memory However, the vastly ncreased memory bandw dth shou d a ev ate the performance h t cons derab y

The Radeon 9700 is one of the first graphics chips to support the new AGP 8X specification Doub ng the bandw dth prov ded by the prev ous specificat on, AGP 8X prov des a 2Gb/sec nk w th main system memory This new AGP channel might look puny compared to the graph cs ch p s own oca memory nterface But as games ncrease in complexity, the ncreased bandwidth offered by the mproved nterface w come in handy

The Radeon 9700 a so has what ATI ca s HyperZ II Catchy market ng term nology as de, there are some interest ng techno ogies wh ch further ncrease performance ntroduced w th the orig na Radeon, HyperZ II is the th rd ncarnation of the Z-buffer cu ing and compress on technology ATI c a ms that t ncreases performance by a most 50 percent

Quality Per Pixel

AT s justif ably proud that the Radeon 9700 s the first graph cs ch p to fu y support Microsoft's upcom ng D rectX 9 API specification Beyond the usua p thora of nteresting new features, DirectX 9 a so marks a s gn f cant sh ft n focus for the PC games ndustry *One of Microsoft's key requirements for full DirectX 9 compatibility is a high precision, purely floating point based pixel shading pipeline Currently, the Radeon 9700 is the only 3D chip in the market which completely meets this requirement*

As mentioned before, w th the introduction of the Radeon 9700, ATI s keen to stress the qua ty of the overa l visua experience They c a m that their new graphics chip s best described as a VPU or V sua Process ng Un t wh ch s "dr v ng the v sual processing exper ence revo ution "

The company ust fies th s hype by emphas s ng that the qua ty of the visua s p oneered by the new Radeon 9700 VPU s on an order of

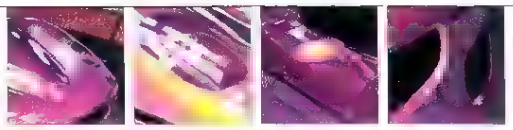


Figure 1 Two tone, suspended microflake-layered car paint rendered in real time on the ATI Radeon 9700

image quality. The final rendered scene might exhibit clearly visible artefacts.

Writing in one of his famous plan file updates in April 2000, Carmack said, "Once I sat down and started writing new renderers, the precision issue has started to bite me personally. There are quite a few times where I have gotten visible banding after a set of passes, or have had to worry about ordering operations to avoid clamping 64 bit pixels. It's The Right Thing to do."

The Radeon 9700 ... granted, John Carmack's plea for improved image quality but actually ... the legendary developer asked for only 64-bits of precision per pixel, the Radeon 9700 delivers twice that quality with its 128-bits per pixel rendering precision. ATI can justifiably claim that their VPU is the only one that will show off next generation ... [illegible]

SmartShader

As mentioned previously, the Radeon 9700 ... to fully support all the DirectX 9 features ... generation texture ... [illegible] ... Each ... 64 000 ... *processing power available in any PC ... until now.*

The VPU's pixel shader capabilities are equally impressive. The Radeon 9700 boasts enhanced internal and external precision, enabling full 128 bits per pixel colour accuracy. This includes a guaranteed 96 bits per pixel precision for all pixel shader operations. In addition to DirectX 9, the new pixel shader functionality will also be exposed via ...

Here's an example of an impressive shader effect which is possible only by using the capabilities of the Radeon 9700 and DirectX ...

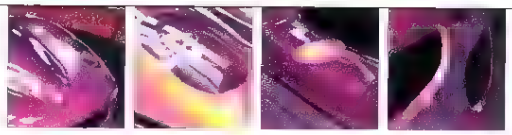
Two-tone, suspended microflake-layered car paint shader

This is an interesting pixel shader ... simulating two tone, suspended metal flake car paint. The car model shown ... relatively modest number of polygons, but uses a normal map generated from ... appearance preserving simplification algorithm ...

Visit http://www.ati.com/developer for more ...

... than a traditional 8-8-8-8 normal map ... 10 or 11 bits per channel is not enough ...

The normal decompression described above is performed on a surface normal map which is generated from an appearance preserving ...



simplification process (Ns) and a high frequency normalized vector noise map (Nn) which is repeated across the surface. These two normals are used to compute two perturbed normals which are then used to simulate the two-toned nature of the paint as well as the reflectance suspended in an inner coat of the paint.

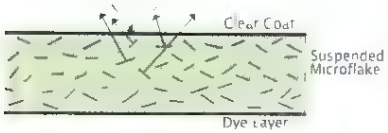


Figure 2 Metallic microflakes suspended in clear enamel is applied over a base paint coat (dye layer) and results in subsurface light scattering.

The perturbed normals Nv and Nw are computed as follows:

Nv = aNn + bNs, where a << b

Nw = cNn + dNs, where c << d

The coefficients a, b, c and d above are constant input parameters to the pixel shader which essentially determine the distributions of the perturbed normals which in turn determine the width of the region in which the microflake is readily visible. These two normals are dotted with the view vector and used as parameters in the following polynomial, which determines the color of the base coat and strength of the microflake term:

c0(Ns·V) + c1(Ns·V)2 + c2(Ns·V)4 + c3(Ns·V)16

Essentially, the first three terms of this polynomial perform the blend between the two tones of the paint and the fourth adds an extra layer of sparkling from the microflake.

The final step in rendering the painted areas of the car is the inclusion of the clear coat through the addition of an environment map as shown below.

One interesting aspect of the clear coat term is the decision to store the environment map in an RGBScale format to accumulate high dynamic range in a low memory footprint. The alpha channel of the texture shown on the right below, represents 1/16th of the true range of the data while the RGB, shown on the left below, represents the normalized color. In the pixel shader, the alpha channel and RGB channels are multiplied together

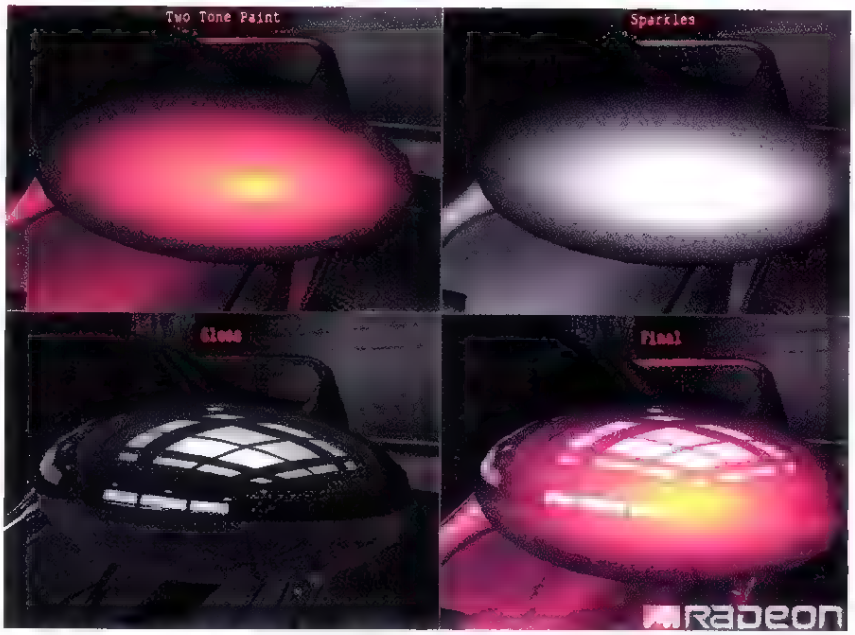


Figure 3 Two tone, microflake, clear coat and final lighting on side rear view mirror

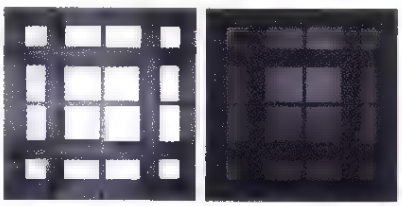


Figure 4 RGB channels of top face of HDR cube environment map

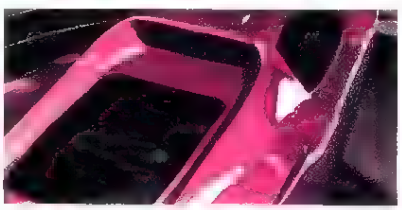


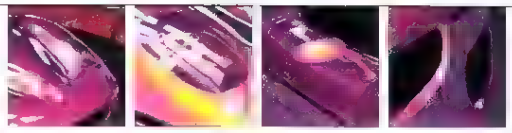
Figure 5 Rough specular black paint and other trim

and multiplied by eight to reconstruct a cheap form of HDR reflectance. This is multiplied by a subtle Fresnel term before being added to the lighting terms described above.

Trim areas of the car such as the black plastic area on the back and the grooves around the door are separated out by a monochrome base map. The monochrome map knocks out the metallic shading for those pixels of the car surface and also determines a per pixel LOD bias to use when accessing the environment map. This allows us to cheaply simulate more rough surfaces in trim areas. The colour from the environment map was converted to a luminance value in the pixel shader as it looks better.

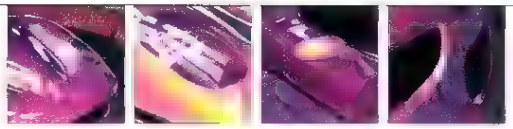
The full pixel shader for the car paint and trim is shown below.

```
PsConst 3 (-4.0, 4.0, 12.9, 0.50)

// sparkle strength, sparkle
PsConst 4 (0.10, 1.00, 16.0, 0.0)

PsConst 9 (0.500, 1.00, 1.00, 0.00)

PsConst 5 (0.4, 0.0, 0.35, 0.0)
PsConst 6 (0.6, 0.0, 0.00, 0.0)
PsConst 7 (0.0, 0.35, 0.35, 0.0)
PsConst 8 (0.5, 0.5, 0.0, 0.0)

ps.2.0

def c0, 0.0, 0.5, 1.0, 2.0
def c1, 1.00, 0.00, 1.00, 0.00

dcl_2d s0
dcl_2d s1
dcl_cube s2
dcl_2d s3
dcl_2d s4

dcl t0
dcl t1
```



Interview With Michael Smith, Developer Relations Manager

Now that the Radeon 9700 has been launched, what are the challenges that you face in educating developers about its potential?



Would you agree that PC games software is lagging behind the hardware by two to three years? What can ATI do to improve the situation?

t s diff cult for games software to keep up with short hardware re ease cyc es Games are on a 12-18 month development cyc e, while hardware s being cranked out every 6 months It d d take a ong t me for deve opers to get good DirectX 8 shader support out

W th D rectX 9, you can go from the 1 4 pixel shader model to the D rectX 9 model fa rly easy You just have to make sure your app cation compi es under D rectX 9 and make some sma changes after that to support 2 0 shaders

ATI s assisting SV's w th the development of 2.0 shader code and mplementation RenderMonkey and our Software Deve oper's K t are a so he ping to shorten game development cyc es and promote the use of exc ting new effects poss b e with DirectX 9

Of late, ATI's devrel team seems to have been infused with a new vitality. What has brought about this change?

Ha v ng t e fastest graph cs product on the p anet doesn't hurt ATI has a ways had great products, but the RADEON 9700 s ser ous y pushing the ead ng edge of graphics techno ogy a ow ng extremely h gh rea sm n rea t me

ATI s backing of open standards such as DirectX 9 H gh Leve Shad ng Language and OpenG GL2 has been widely accepted by the development community Developers are very exc ted about the future and are us ng the 9700 as the dev e opment ve h c e for t e r current and next gener at on ap cat ons We have expanded our deve oper support teams n both North America and Europe to meet demand and ensure ISV's have the tools and nformation necessary to get the most out of ATI products mov ng forward

So, what do the games deve opers make of the Radeon 9700 s features? We talked to a few of them to find out

Core Design

After storming the gam ng market e y Chr stmas with the r bouncy pony ta e t hero ne, Lara's creator has now gone back t the draw ng board The next ed t on of Miss Croft's adventures w be powered by a brand new engine, sport ng the latest D rect X vertex and p xe shader effects

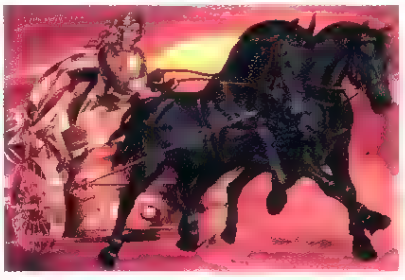
Pixe shader 2 0 is a b g step n the correct d rect on says Duncan Hopk ead Graph cs Progr er at Core Design The Radeon 9700 s t e f rst card to support the texture formats w l ope up a whole world of new techn ques that w a ow a lot of the work done on the vertex shader to be mo ed to the pixe shader, reducing p

Core Design Ltd
Lara Croft Tomb Raider and Core are registered trademarks of Core Design Ltd All rights reserved

Candella Software

A tr bute to the o d Hol ywood epics such as Ben Hur, Candela's deb t rac ng game Char ots The F rst Olympics, uses both pixe and vertex shaders In add t on, the game use s t ee textures per pass Th s s where the Radeon 9700 s h gh internal co our prec s on and f ll-rate come nto the p y

Heavy duty mu t textur ng l ke n Char ots w l always strain mage qua ty" exp ains Rob Grip, the projects 3D graphics programmer "But w th the new Radeons s h gh qua ty internal frame buffers, there are



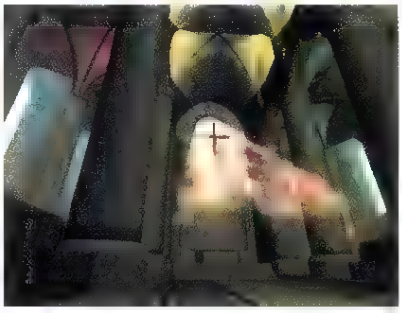
Codemasters Software Company Limited
Colin McRae Rally 2.0 are registered trademarks of Codemasters Software Company Ltd All rights reserved

t race ef ect t certa n y s the future for the ndustry

Funlabs

The Fun abs team s current y work ng on Cr s eed n-house 3D engine wh ch mp ements both vertex and p xe shader They are a so us ng the programmab e p pe ne for anima t ng the skinned models in their upcom ng game The programmab e p pe ne s used for operations such as ve tex sum nation mesh b end ng and for the effects appl ed t the sk ned mode s such as per p xe ight ng Bu n t a of these routines were prev ous y calcu ated on the CPU Th s has a big impact on the frame rate

The use of the programmable p pe ne s not m ted to the render ng of sk nned mode s Other e ements of the engine a so make equa y eff c ent use of the techno ogy For stance some effects such as d storted mirror reflective r pp ng water and shadows a use the D rectX 9 programmab e p pe ne



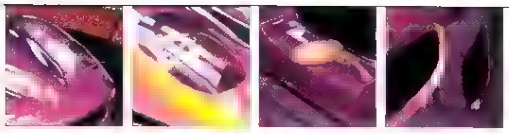
Funlabs
FUNLABS 2002

Mind Your Language

For the software deve oper, one of the ma n novat ons in D rectX 9 s the ntroduct on of the Microsoft H gh Leve Shad ng Language HLSL Simu taneou y, the OpenG ARB is finalising the spec f cat ons of OpenGL 2 0 which nc udes its own HLSL

Currently, games developers have to code a the vertex and p xel shaders in assembly Th s is a tedious and t me consum ng task Wit the la nch of powerfu new hardware l ke the Radeon 9700 PRO, this can be come even tougher f you choose to code in assembly But if you use a HLSL, th s task becomes much more simp e than wr t ng everyth ng in assembly

The examp e be ow c ear y i ustrates th s



Assembly Code

```
ps.2.0
def c0, 2.0f, -1.0f, 0.5f, 0.5f // scale/bias
def c1, 1.0f, -1.0f, 0.1f, 0.0f

dcl_2d s0                       // Perturbed

dcl t3.xyzw                     // xyz == Ps

dcl t4.xyzw                     // xyz == Pst

dcl t6.xyzw                     // xyz == Peye (v
dcl t7.xyzw                     // xyz == Neye (v

texld r7, t4, s0                // Sample dY from
mov r3.y, r4.x                  // Put dY in y
mov r3.z, r5.x                  // Put dZ
mov r6.y, r7
mad r6, r6, c0.x, c0.y          // Put into [-1,1]
mad r3, r3, c0.x, c0.y          // Put noise in [-1,
texld r3, t0, s1                // Sample noise
add r7, c4, -r3                 // Add to Pstad
nrm r4, r7
dp2add r0, r7, r7, c1.w         // 1 - (x^2 + y^
rcp r0, r0.x                    // 1/sqrt(x2 + y^
mul r0, r0, c2.w                // sqrt(x^2 + y^
texld r0, r0, s1                // Sample from 1D n
mov r1, c3
dp3 r2, r0.x, c2, r1            // Perturbed Nor

sub r4, c4, t6                  // Compute normalize
nrm r5, r4                      // Vertex to light in eye space
dp3 r5.w, r4, r4
rsq r5.w, r5.w
mul r4, r4, r5.w                // Leyedp3 r6.w, t7
t7                              // N
r6.w, r6.w
mul r5, t7, r6.w                // Neyedp3 r3.w, t6,
t6                              // Peye
Peye to the power rsq r3.w, r3.w
r3, r3, t6, r3.w                // Peyeadd r6, r3, r5

Peye + Peyedir r6.w, r6, r6
rsq r6.w, r6.w
mul r6, r6, r6.w                // Heyedir sat r6
r5, r6                          // N H
mad r0.z, r0.z, c5.z, c5.w
r6, r6.x, r0.z                  // N H
r5, r4, r5                      // N
r5, r5, c0.z, c0.z              // H
r6, r6, r0.y                    // Gloss term in alpha
texldp texture mad r2, r5, r2, r6
sdec rmov oC0, r2
```

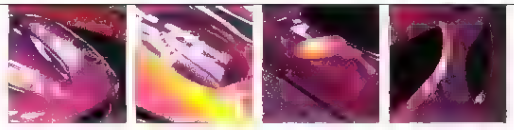
DirectX 9 HLSL Code

```
float4 hlslWood( float3 Pshade0
: TEXCOORD0, float3 Pshade1
: TEXCOORD1, float3 Pshade2
: TEXCOORD2,
                float3 zWobble0 : TEXCOORD3
, zWobble1 : TEXCOORD4, float3 Peye
: TEXCOORD6, float3 Neye : TEXCOORD7
)
{
    coloredNoise,
    wobble
    coloredNoise.x = tex3D(NoiseSampler,
Pshade0); //

    coloredNoise.y = tex3D(NoiseSampler,
Pshade1
    coloredNoise.z = tex3D(NoiseSampler,
Pshade2,,
    wobble.x = tex3D(NoiseSampler,
zWobble0,
    wobble.y = tex3D(NoiseSampler,
zWobble1)
    wobble.z = 0.5f,
    coloredNoise = coloredNoise * 2.0f - 1.0f
    wobble = wobble * 2.0f - 1.0f

    float3 noisyWobblyPshade = Pshade0 +
coloredNoise * psConst3.w + wobble *
psConst4.w
    float scaledDistFromZAxis =
sqrt(dot(noisyWobblyPshade.xy,
noisyWobblyPshade.xy)) * psConst2.w,
    float4 blendFactor = tex2D
(PulseTrainSampler, float2(0.0f,
scaledDistFromZAxis),
    albedo = psConst2 * blendFactor.x
+ psConst3 * (1 - blendFactor.x),

    float3 Leye = (psConst4 - Peye) /
length(psConst4 - Peye),
    Neye = Neye / length(Neye),

    float3 Veye = -Peye / length(Peye),

    float3 Heye = (Leye + Veye) / length(Leye +
Veye),,
    float NdotH = clamp(dot(Neye, Heye),
0.0f, 1.0f),
    float k = blendFactor.z,

    float specular = tex2D
(VariableSpecularSampler, float2(NdotH, k))

    float Ndotl = dot(Neye, Leye)

    float diffuse = Ndotl * 0.5f + 0.5f

    float gloss = blendFactor.y

    return diffuse * float4(albedo.r, albedo.g
, albedo.b, 0.0f) + specular * gloss,
}
```

It should be clear from this example that the HLSL version is much more readable and maintainable. It will also run just as fast. With the current beta release of the DirectX 9 HLSL compiler, the HLSL version is only 2 ALU instructions longer than the hand-coded assembly version. This is an important point and says a lot about the effort Microsoft is putting into optimising the compiler and language.

While choosing a higher level shading language is important for games developers to adopt a truly open standard. Currently PC graphics hardware of varying capabilities are available from multiple vendors including ATI, Nvidia, SiS, VIA-S3 Graphics, Intel, Matrox, Imagination Technologies, Trident Microsystems and 3DLabs.

Only an industry-standard high level shading language that is part of DirectX 9 or OpenGL 2.0 is guaranteed to be compatible with this wide variety of graphics hardware. Some proprietary high level languages like [illegible] designed and promoted by ATI's main rival, Nvidia, do not carry the twin crucial promises of compatibility and performance across all graphics hardware.



Nvidia had recently claimed that Cg is fully compatible with the DirectX 9 HLSL. However, Ken Schramm, spokesperson for the DirectX group has a different opinion. 'Nvidia's Cg is not the same as Microsoft's high level shader language as each are developed by separate companies.' he emphasizes. *"As Nvidia is free to diverge from our standard, Microsoft cannot guarantee 100% compatibility going forward."*

CodePlay's Andrew Richards is similarly pessimistic about Cg. "Overall, Cg is not a generic language for graphics programming, it is a language for Nvidia's current generation of graphics card." he points out. "It has a different set of goals from what's required to design computer graphics at the heart of the successful computer games of tomorrow."

Quite recently at SIGGRAPH, even Nvidia conceded that Cg is optimized only for their own hardware. Obviously, this completely rules out its use by developers who want to see their game run on all graphics hardware.

RenderMonkey

Instead of developing a proprietary language like Cg, ATI has instead chosen to focus their efforts on developing utilities built on industry standards like the DirectX 9 HLSL and OpenGL 2.0. This guarantees that developers do not have to worry about compatibility with the different graphics hardware in the market.

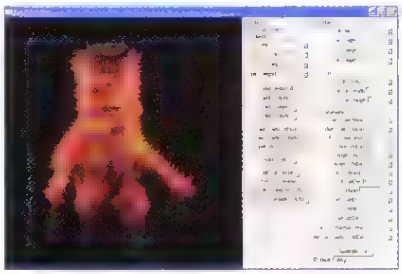
Spearheading this new suite of utilities is RenderMonkey, an intuitive GUI-based integrated development environment. RenderMonkey was created with three main objectives in mind.

Firstly, ATI wanted to provide a tool with a MS Visual Studio-like interface which would already be very familiar to the programmers. It would allow the creation, editing and real-time visualisation of vertex and pixel shaders. It would also allow developers to take these shaders and use them in their own applications.

The second problem that ATI wants to overcome is how shaders are generally used in a games development environment. Shaders created by programmers end up being used by artists. *Since most artists have very little programming knowledge, ATI wanted to provide them with a tool which would allow them to tweak the shader variables without having to depend on the programmers.*

And finally, ATI wanted to create a tool that was extensible and could be integrated easily with the existing in-house tools. The entire RenderMonkey utility is built using XML, the universally accepted markup language. This allows each development house to add their own extensions to the utility.

To give an example of how it can be used by non technical artists, take a look at the interface of the fur generator tool that is available as part of the RenderMonkey tool suite. This tool provides a host of variables which the artists can tweak to get the exact look that they need. And all this is possible without the need for supervision from the graphics programmers.



Interview with Callan McInally, Manager, 3D Application Research Group

Why should games developers use RenderMonkey when the same effects can be achieved using a higher level shading language like the one included in DirectX 9?

The DirectX 9 High Level Shading Language provides a better way of writing the low level assembly code. However it does not solve the same developer problems for which we designed RenderMonkey.

RenderMonkey was designed to support easy encapsulation of all the parameters needed for an effect - not just the assembly code - and to provide a workflow tool to aid developers in creating, managing and using these effects. The HLSL should be viewed as complementary to RenderMonkey rather than competing with it. The DirectX 9 version of RenderMonkey has complete support for both HLSL and assembly shaders.

Do you wish to see developers integrating RenderMonkey into their own in-house game engines or editors?

Yes. In fact, the number one request we have had from developers since we released RenderMonkey is for the plugin SDK so they can tightly integrate RenderMonkey with in their existing workflow.

We designed RenderMonkey from the start to be as extensible, open and flexible as possible to allow developers to use it at the level that is best for them. Some developers have already developed in-house tools that complement RenderMonkey. They are happy to simply use RenderMonkey as a prototyping tool. Other developers who haven't yet developed proprietary tool sets are taking the decision to use RenderMonkey as the core component of their entire shader tool set.

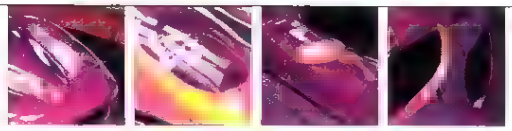
Is the output shader code of RenderMonkey useful just for ATI graphics cards or is it compatible with all graphics cards in the market?

It is compatible with all graphics cards in the market that have support for DirectX 8.0 or higher. I don't believe it is in our best interests to be competing with the other graphics card vendors for developer mind share.

Developers have limited resources for adding new graphics features into their titles. Proprietary solutions only further dilute this limited resource. The net result is that fewer titles appear on the shelves with the latest graphics features. Hence, you end up with poorer quality graphics and a less impressive experience for the consumer.

What are your future plans for RenderMonkey?

Our focus for the short to mid term is on adding support for the emerging High Level Shading Languages. This will include the DirectX 9 HLSL, the OpenGL 2.0 Shading Language and any other interesting languages. I can't see the compiler for a subset of the RenderMan shaders that we showed at SIGGRAPH this year.



Tighter integration with the major digital content creation packages is also on the agenda. We've provided a good artist's interface in RenderMonkey to [illegible] programmers & artists to collaborate creating shaders. However the artists who simply wish to apply and tweak shader parameters would prefer to do this in the applications they are most familiar with.

We also aim to provide more documentation, more tutorials and more advanced examples to ensure that all the great DirectX 9 shaders we developed for the Radeon 9700 are provided to developers as soon as possible so that they can reuse them in their own titles.

We'll also be extending RenderMonkey with a number of custom plugin components. We already have some of these in development such as the Fur Generator plugin and a component for generating procedural noise textures. We also plan on providing import/export support to all of the major engines.

We're open to any and all suggestions on how to improve RenderMonkey. We developed this tool to be useful for [illegible]. Only they can tell us if we have succeeded and what we can do to make RenderMonkey even more useful. We expect around 70% of the [illegible] features that we add to RenderMonkey in the future will come from suggestions by external developers.

FireGL X1: A Paradigm Shift Towards Real-Time Cinematic Rendering

Real-time rendering of cinematic content has been one of the holy grails of all graphics chip manufacturers. At present all special effects in a popular TV series like Star Trek or a movie like Star Wars are rendered non real-time using clusters of interlinked server [illegible] at a highly expensive render farm.

A single frame in non real time rendering could take several minutes or even hours to render. Obviously this is a far cry from the typical PC or console gaming environment where players are used to dozens of frames per second.

[illegible] Courtesy of Rhythm & Hues Studios.

Until recently, it had been thought that offline cinematic rendering like that used in Hollywood would not be able to reach real-time frame rates for several years. Even taking into account the CPU performance increases delivered by Moore's Law it would take several years, if not decades to achieve real-time frame rates.

Conversely many in the movie industry had scoffed at the idea of ever being able to use consumer level graphics cards to accelerate movie rendering times as they simply did not have the internal precision and quality of output required for the big screen.

All this changed with two major developments.

First came the publication of a seminal paper at SIGGRAPH, 'Interactive Multi-Pass Programmable Shading', by Mark S. Peercy, Mark Peercy, now an ATI employee clearly showed in his paper that it was possible for OpenGL-enabled consumer level graphics chips to render cinematic quality images at almost real-time frame rates.

Peercy showed that even complex offline shader code like those used in RenderMan could be broken down into multiple smaller steps which would then be rendered real-time on the VPU. Since a VPU like the Radeon 9700 has an extremely high fillrate, it is possible to render the offline images at very close to real time frame rates.

The second crucial development has been the launch of the Radeon 9700 VPU itself, the first consumer level graphics chip with full internal floating point precision.

ATI's workstation board, the FireGL X1 uses the Radeon 9700 VPU.

Using the methodology espoused by Mark Peercy and the floating point rendering precision of the FireGL X1 it is now possible to render movie quality images at close to real time frame rates.

As usual, [illegible] have the last word on this.

Writing in a Slashdot forum he says: 'Peercy's (IMHO seminal) paper showed that given dependent texture reads and floating point pixels you can implement RenderMan shaders on real time rendering hardware by decomposing it into lots of passes it may take hundreds of rendering passes in some cases meaning that it won't be real time but it can be done and will be vastly faster than doing it all in software. It doesn't do [illegible] absolutely everything [illegible] but it [illegible] gives [illegible] magnitude improvement in price performance and cycle time [illegible] specify, say, the exact filter kernel [illegible]'.

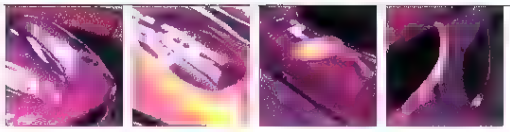
There will always be some market for the finest possible rendering, using ray tracing, global illumination etc in a software renderer. This is analogous to the remaining market [illegible] for supercomputers [illegible] some applications, it is still the right thing [illegible] can afford it. The bulk of the [illegible] migrate to the cheaper platforms.

[illegible] *developing*, that will go from existing languages instead of sending their RIB code to the renderfarm, you [illegible] program that decomposes it for hardware [illegible]. They will return image [illegible] like everyone is used to.

[illegible] Courtesy of Rhythm & Hues Studios.

Multi chip and multi card solutions are also coming, meaning that you [illegible] more frame rendering power in a single tower case than Pixar's entire render [illegible] Next year.

[illegible] had originally estimated [illegible] a few years for the tools to mature to the point that they would actually be used in production work. But some companies have done some very smart things and I expect that production frames will be rendered on



PC graphics cards before the end of next year. It will be for TV first, but it will show up in film eventually."

The 'transistors, multi-chip and multi-card solutions hinted at by Carmack will all soon be available on ATI's FireGL X1 platform. Very soon, the long-held dream of the merger of real-time and off-line rendering will be realised.

Interview with Raja Koduri, Applied Engineering Manager

Could you briefly explain how the FireGL X1 is a major step forward in closing the gap between real-time rendering and off-line rendering?

The FireGL X1, enables for the first time ever, shading descriptions used in off-line rendering such as the RenderMan Shading Language, Maya Shading DAGs, and 3D Studio Max materials to be compiled directly for rendering on the graphics hardware rather than the CPU.

The enabling technology in this paradigm shift is floating point. Floating point pixel operations, floating point textures, and floating point frame buffers. The Radeon 9700 chip, on which the FireGL X1 is based, is the first and currently, the only consumer graphics chip to support floating point operations throughout the graphics pipeline.

Floating point is required because the shading descriptions used in off-line rendering demand native floating point data and calculations. Although game artists have used incredible talent to create stunning images in fixed point for real-time rendering, they have always been constrained in their shading by the graphics pipeline itself.

The FireGL X1 now releases them from these constraints. Therefore we will see visual imagery which is not even an option on other graphics platforms.

Are you going to introduce any new utilities which will help to accelerate digital content creation?

We are developing a number of demonstrations based on the principle of compiling shading descriptions used in off-line rendering. These include a compiler for the RenderMan Shading Language, a compiler for the Maya Shading DAG, and a compiler for 3D Studio Max standard materials.

We are developing a 3D Studio Max plugin that embeds the Max material compiler in a runtime environment. In this way, we are able to render Max standard materials on Max object geometry directly in a Max viewport. It can be thought of as a hardware preview window that replicates the same images from a shading standpoint as the software raytracer.

We currently are well into development of the RenderMan Shading Language path, as well as the 3D Studio Max path. The Maya path will be coming [illegible]. The goal of these demonstrations is to enable our partners and developers to improve their art production pipelines through improved tools and higher artistic throughput.

We understand that every studio has unique workflows and do not expect, in any sense, to replace their current tool sets. Therefore these demonstrations will not necessarily implement the entire functionality exposed in RenderMan, Max, or Maya. Rather, we will be making all of the demonstrations freely available so they may be extended and adapted to meet each user's need.

Can you mention any movie effects studios who are currently using the FireGL X1 and how they are using it.

Quite a few DCC houses around the world are currently using or evaluating the FireGL X1. For instance, Weta Digital who created the special effects in Lord Of The Rings is using dozens of our FireGL boards in their modeling and lighting departments.

"Sum of all Fears" © 2002 Paramount Pictures
Courtesy of Rhythm & Hues Studios

Interview with Richard Huddy, Managing Director, Code Mafia

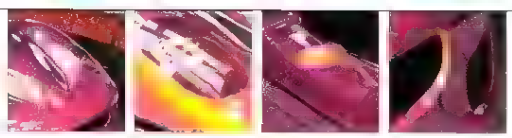
Before you formed Code Mafia, you were part of Nvidia's developer relations team. So, why did you choose to work with ATI instead of Nvidia?

When we started Code Mafia it was because we really wanted to be part of the solution, rather than part of the problem. We wanted to be able to focus on solving developer's problems while still using our expertise in PC and Xbox graphics. That's still the "mission statement" for Code Mafia, and it turns out that this simple-to-state goal is shared by ATI as well.

Since the moment ATI acquired ArtX, there has been a real change in their attitude. ATI is keen to compete in the PC graphics business on the basis of pure technology and open standards and the Radeon 9700 is a straightforward example of awesome technology.

It's not just faster than anything else available but it's also easily the most fully featured board that money can buy. ATI is working really hard to make sure that all these new features are available to every developer. Their board seeding programme is huge and rapid, their demo team is cranking out new demos with really practical uses at an astonishing rate and their commitment to the twin open standards of DirectX 9 and OpenGL marks them out as a real leader.

Everything I see about ATI says that they share the same passion and conviction which drove me and my colleagues to form the Code Mafia. Graphics innovation is 'important', but innovation is more than just important – it's essential to a thriving games industry. And I'm confident that ATI has the ability to make a really valuable and lasting contribution to the games industry.



How do you see the ATI Radeon 9700 being used by PC developers in the immediate future?

Right now it's the platform of choice for two reasons. I guess the obvious reason is that it's much faster than anything else you can lay your hands on. But speed isn't the only thing that the Radeon 9700 is about.

The long term advantage of the Radeon 9700 is that it's a fully featured DirectX 9 graphics card. With the Radeon 9700, you gain access to a swathe of new features which show the way to the bright new future of PC graphics. Pixel shaders 2.0 and vertex shaders 2.0, a fully floating point pipeline from start to finish, a 256-bit memory bus that delivers over 20Gb per second of local memory bandwidth and a massive 128MB frame buffer which allows you the space to code up new techniques and to experiment with new strategies of handling graphics problems.

It's the single best development board available, and it'll remain right on the leading edge for about a year. And it's also just what artists are looking for. With that much raw power and performance, it's the ideal environment for running all the heavyweight Digital Content Creation applications.

Is it really possible for a developer to design a game for both DirectX 8 and DirectX 9 without the need to have multiple code paths in the renderer?

It turns out that this kind of a goal of having a single common code path for all graphics development is just impractical.

The practical consideration is the toughest one. Most PC graphics programmers will tell you that coping with compatibility is one of the hardest parts of their work - and because of that struggle they end up with several code path for the many different graphics cards anyway.

Sometimes it's so that they can work around bugs in specific drivers or hardware and sometimes it's because that's just the best way to handle one type of graphics card which just doesn't suit all the others. For that reason, code paths in PC graphics tend to proliferate rapidly.

But Microsoft does a cracking job of handling the bulk of backward compatibility and that's why any DirectX 9 game can be written in such a way as to run on any hardware which has a DX7 driver. So you can imagine a game player with a piece of DirectX 6 hardware - like say, an ATI Rage 128 - loading up a DX9 game and it will all just work.

The consumer seems a smooth process - they load the game (which maybe installs DirectX 9 for them) they run the game and it works. That makes it all look simple and fun. Of course underneath the hood, there's an awful lot going on to make that happen and some of that is done by the game developer, some by the DirectX runtime. But the end result is pretty impressive.

And if the consumer buys a new graphics card and plugs it in, then there's a good chance the game will just run faster and look better than before.

What are your key goals at Code Mafia? How would you be able to help PC games developers?

I want to be involved in some part of the games development process which helps developers rather than holding them back. I want to be able to share the broad experience that we've built up in this industry and so make the process more about fun and less about pain. I want to work in an industry full of smart people doing interesting things and solving difficult problems.

And, of course, I share all these desires with my colleagues at the Code Mafia and at ATI. That's why we've chosen to work together. But then, we don't think that this is too much to ask. We love this industry and we're grateful for the opportunity that ATI has put in front of us.

Richard Huddy's DirectX Programming Tips

Sort by vertex shader or sort by texture and sort front to back.

Use Clear once per frame and always clear colour, Z and stencil together unless you can't just clear Z or stencil.

Use the standard DirectX 8 vertex buffer handling algorithm with NOOVERWRITE.

Always use DISCARD at the start of a frame on dynamic vertex buffers.

Treat index buffers exactly as vertex buffers except that you always choose the smallest element possible.

Use 16-bit indices wherever you can and 32-bit indices if you only have to.

Keep index buffers out of system memory as they are always treated as 'first class citizens'.

The most expensive state changes are switching between vertex shader and fixed function pipelines, switching vertex shaders and switching textures.

Create your most important resources first – shaders, textures, vertex buffers, index buffers etc.

Avoid all output registers of the vertex shader.

Use the write masks aggressively and pack constants as much as possible.

Branches and conditionals in vertex shaders are fast. So use them aggressively.

With pixel shaders, you now have to explicitly write to oC0, oC1, oC2 and oC3 to set the output colour. Spend the clock cycles on shader operations than texture lookups as memory latency is a killer.

The War Without End

The graphics chip market is a ruthless one. It could even be characterised as war by other means. In this war, the best that any company can hope for is to be the victor for a few months at a time. As many experts have found out, predicting the future of this industry can be an embarrassing experience.

Despite the success of the Radeon 9700 VPU, ATI is not resting on their laurels. Already plans are afoot for the launch of its successor early next year. ATI has also announced that future products will use cutting edge memory technologies like DDRII and GDDRII. All this points to a company that is keen to maintain its leadership position for a long time to come.

Radeon 9700

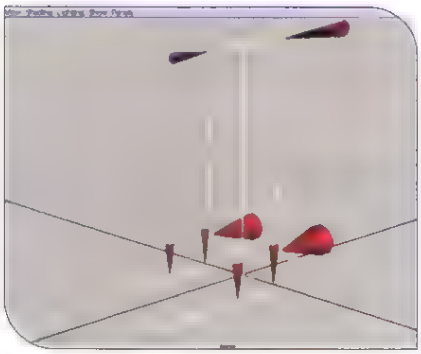


FIGURE 3

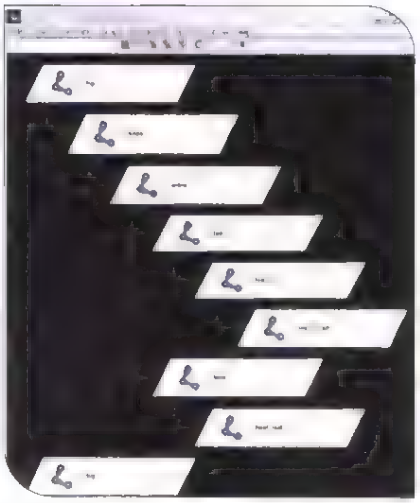


FIGURE 4

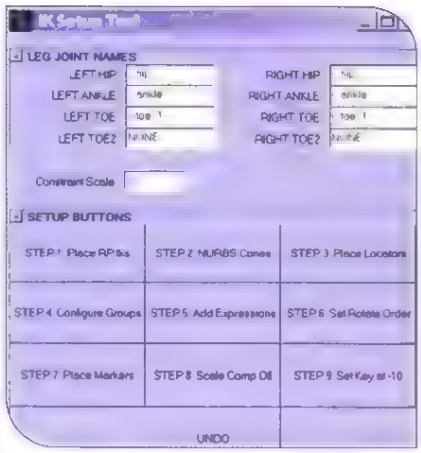


FIGURE 6

based loosely on a character's size. The artist then hit nine buttons in sequence. These buttons would auto-attach the IK handles and instantly build the constraint hierarchy.

Dissecting the IK Setup Tool

MEL is a quirky and often inconsistent language. A good portion of the time we spent developing our IK Setup Tool was used to track down the proper commands for the tasks we needed to execute. Still, we managed to uncover the MEL commands we needed to actuate the core tasks of each of our nine tool buttons.

The first button's purpose was to place IK handles on a character's legs. It read the names of the bones from the top text fields by using the `textFieldGrp` command in its query (`-q`) mode. These string variables were then passed to the `ikHandle` command, which in turn created the IK handles.

The second button placed NURBS cones on a character's hip, ankle, and toe joints. These cones, created using MEL's `cone` command, were the primary constraint objects an animator would use to manipulate the legs. The `xform` command was used to query (`-q`) the positions of the leg bones and store them as variables. The `move` command then read these variables and moved the cones into place. Finally, MEL's `pointConstraint` locked the hip cones to the character's hips.

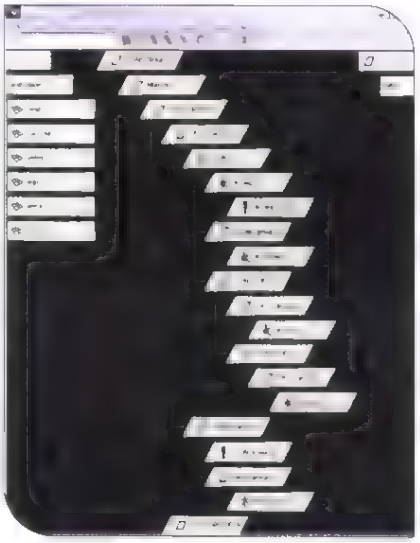


FIGURE 5

Pressing the third button called `CreateLocator` to place a pair of locators in the scene. Next, the `group` command grouped the locators to themselves. Then `xform (-q)` queried the positions of the character's knees, and `move` translated the two new parent objects to the knee joints and the locators to positions in front of the knees.

Button number four configured the cones, locators, IK handles, parent groups, and constraints into a standardized hierarchy via the `parent` command. Again, the new groups were translated into place using `move`. New constraint relationships were created between the knee locators and main leg IK handles, and the new constraint hierarchy and the skeleton. These were implemented using the `poleVectorConstraint` and `scaleConstraint` commands, respectively.

Button five added several expressions to the scene, saving us data-entry drudgery. We added expression code for specifying both constraint and skeletal behavior using the `expression` command, allowing us to automate both the creation and the specifications of our setup expressions.

Number six altered the rotate order of the heel and toe NURBS cones from XYZ to YXZ using `setAttr`. We had previously determined that this rotate order produced the most reliable rotations in our quaint Z-up environment.

Buttons seven through nine performed some final housekeeping tasks. Button seven grouped custom rotation guides to

FIGURES 7A–C. Joint scaling and translation offered an mators direct manipulation of poses and gave characters moves extra verve for the modest cost of a low-tech solution

a character's spine using the polyCube and parent commands. Button eight used setAttr to ensure Maya's segment scale compensate was switched off for all of a character's joints. Finally, button nine keyed a reference frame at –10 on the character's skeleton and constraint hierarchies using setKeyframe. Listing 1 shows some of the MEL procedures we found most useful.

Automating this process with MEL both saved us time and eliminated the steps most prone to human error. Furthermore, by enabling any artist, regardless of their setup experience, to fit a prototype and/or character with a functioning IK system quickly, we alleviated bottlenecks. This conservation of both time and human resources saved energy that could then be devoted to artwork.

LISTING 1. Some helpful MEL procedures

```
// A method for querying a bone's position in world space:
   xform -query -worldSpace -translation my_joint_name;

// A method for querying the contents of a text field:
   textFieldGrp -query -text my_text_field_name;

// A method for setting a keyframe at frame -10 on a hierarchy:
   setKeyframe -time -10 -hierarchy below my_hierarchy_name;

// Basic transformation methods: translation, rotation, and scaling:

      // Moves an object to (0,0,5):
      move -absolute 0 0 5 my_object_name;

      // Rotates an object by 90 degrees on Z,
      // relative to its current Rotation:
      rotate -relative 0 0 90 my_object_name;

      // Scales an object to 3 times its current size:
      scale -relative 3 3 3 my_object_name;

// (Note: All flags are listed in their long forms.)
```

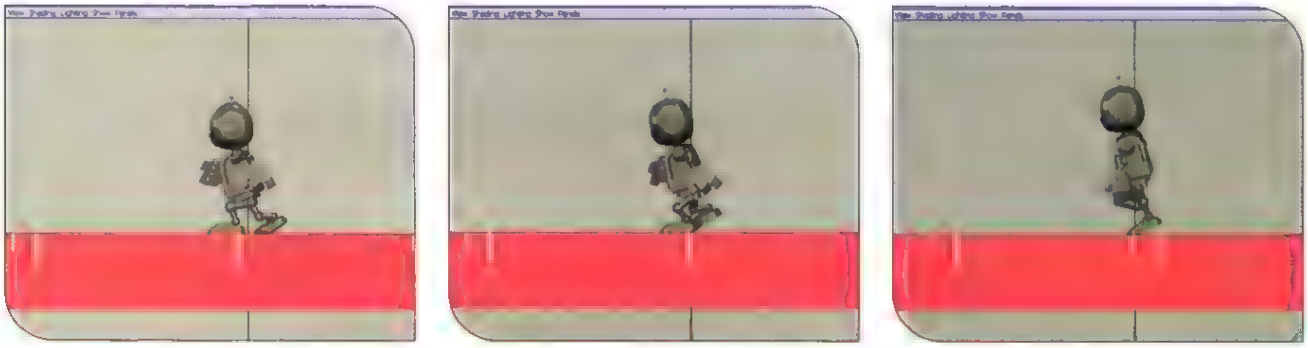
Low-Tech Animation Solutions

The shortcuts and prototypes I've described so far shared a common purpose: to help us create better animation more efficiently. Both of these methods accomplished this by either telegraphing problems or by saving time. Often, however, we would spurn a high-tech solution due to its specificity, inefficiency, and/or complexity. And still at other times, we embraced traditional CG taboos.

We consistently and repeatedly translated and scaled our characters' bones. True, most of us learned on our grandmothers' knees never to do that to a CG character. "Use your constraints," she would say. "Rotate your bones if you must. But avoid scaling them, and don't ever, *ever* let me catch you in a translation!" We all love our grandmothers, but we found that the tenets of traditional animation called for us to — nay, demanded that we — defy her.

The reason behind our disobedience was squash and stretch. We found that by scaling our joints, and especially by translating them, we could instill our animations with extra gravity and snap. Major translations often lasted only a couple of frames and, borrowing an idea from Disney, were "more felt than seen."

Since we had no IK setups to speak of on the spines and arms of our characters, translating the bones in these body parts was quite simple. If needed, we could key the leg IK solvers "off" in order to



FIGURES 8 A–C The Walk Guide helped line up characters' feet on the ground properly every frame to minimize unattractive foot sliding

manipulate these joints. Translation and scaling were effective across the board and worked wonders on anything from walks to attacks to facial animation (Figures 7a–c).

Requiring no additional setup, these low-tech solutions saved us time. Within limits, this method of animation provided animators with a direct, tactile, and expedient method of sculpting their characters' poses. Although unglamorous, this technique was as effective as any in terms of preserving our resources and improving our animations.

Walks and the Walk Guide

Another device we used to aid our animation was called the Walk Guide. We used this tool to help our characters' feet stick to the ground during walk and run animations. Although foot slippage is commonly forgiven in the world of games, we hoped that by eliminating it we could add an extra dimension of believability to our characters' locomotion.

The Walk Guide was an elongated cube with many smaller cuboids attached to it. The smaller cuboids were identical to the polygonal markers on our characters' ankles and toes, which were grouped to their feet during setup.

By scaling a special parent node, the Guide's small cuboids could be adjusted to match a character's foot size. Scaling the large cuboid allowed an animator to accommodate for the character's stride length. A set of constraints and locators ensured that as the stride length changed, the preset foot size remained constant.

Since our walk cycles were animated in place, we needed a way in which to simulate forward movement while keeping track of the positions of a character's feet. The solution was to animate the Walk Guide to the speed specified by the designer (2 meters per second, for example). Once the Walk Guide was moving at the proper speed and the small cuboids correctly scaled, an animator could begin working on the character's walk cycle.

The trick to using the Walk Guide to eliminate foot sliding was to keep the character's foot markers lined up with the small cuboids on the Guide. This applied for every frame in which the foot made contact with the ground (Figures 8a–c).

Upon a cycle's completion, a character could be put into a level and moved at its preset speed with little or no foot slippage. Additionally, programmers could scale the playback speed of the cycle relative to the character's velocity and still have the feet stay grounded.

There were several gameplay situations that were not as clean as the test case I just described; however, the Walk Guide did serve to plant our character's feet properly in most of our worlds. Once accustomed to the Guide, we animators found that using it benefited both our schedule and our artwork, as it kept track of the more technical aspects of locomotion for us.

Making Faces: Artistic Reasons and Technical Details

We knew from the start of developing RATCHET & CLANK that facial expression would be an important component not just to our cinematics but to our gameplay animations as well. Once again, we were faced with the dueling goals of animation depth and scheduling efficiency. We settled on two methods for making faces: a simple one for our enemies and one more complex for our heroes. Expressions exaggerated the idles, intensified the attacks, and sealed the deaths of our enemies and heroes alike.

When animating our enemies, we drew on a traditional animation dictum: A viewer of animation is usually drawn to a character's face, particularly to the eyes. Attention paid to a character's eyes and mouth was very important to making convincing actions, especially during our quick gameplay cycles.

Most enemy characters had fairly simple face skeletons. However, these skeletons allowed for a high degree of manipulation of the eyes and mouth. Each eye had between two and four bones controlling its brow and lids. Mouths were generally simpler, using only one or two bones. In most cases, this setup gave us all the flexibility we needed to exaggerate the enemy's features and thus heighten the emotion of its actions (Figures 9a and 9b).

Our heroes' faces had a more sophisti-

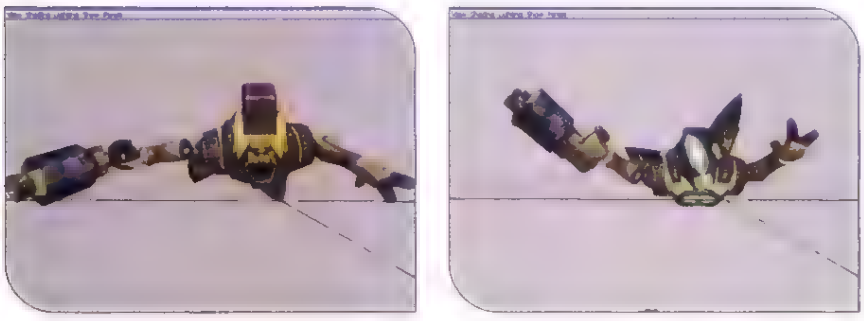
cated setup, which they shared with the NPCs. Though NPC faces were manipulated mostly in our cinematics, Ratchet and Clank made heavy use of expression during gameplay, as well.

Like the enemy setups, hero and NPC faces were manipulated via their face joints. Unlike the enemies', these joints were animated though a driven key system instead of being transformed directly. Since they clocked more screen time, hero and NPC faces tended to have a far greater amount of bones — and hence expressive range — than their enemy counterparts.

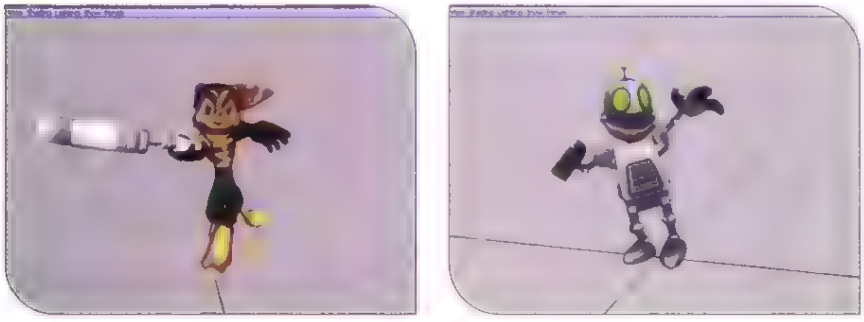
Figures 10a and 10b show some of the range of expression Ratchet and Clank exhibit during gameplay. Ratchet smiles when excited, grimaces when he's hit, grits his teeth during combat, chatters them when he's cold, and drops his jaw when he dies. Clank's expressions change both while he's strapped to Ratchet's back and when he's played independently.

As I mentioned earlier, hero and NPC expressions were animated by combining preset driven key attributes via a MEL script slider interface. These presets allowed the animator to combine and create a wide array of facial expression without having to build them from scratch. Like color primaries, these attributes could be blended together to form new combinations.

About half of a character's 40 or so facial attributes were dedicated to producing a basic expression, either on all or on parts of the face. These basic expressions included anger, disgust, fear, happiness, sadness, and surprise, all of which would be easily recognizable to a player. More subtle attributes were dedicated to animating phonemes and controlling individual facial features. Unique and varied emotional ranges could then be achieved by combining expression, phoneme, and feature attributes together.



FIGURES 9A & 9B. With enemy face skeletons, less was more. Bone detail was reserved for the eyes and mouth to enable simple, exaggerated expressions. Here, during an in-game animation, the Robot Paratrooper's face reacts to being knocked down.



FIGURES 10A & 10B. Ratchet and Clank's gameplay facial animation system (also used for NPCs) needed more sophisticated setups than enemies' for the broader range of expression they were required to show during gameplay.

Scripting Facial Presets

Assigning facial presets to our characters cost us some setup time. However, we were able optimize some of the processes with another MEL script. Like our other MEL tools, this script automated some of the tedious steps, allowing a setup artist to spend more time on the art of sculpting facial poses.

Facial presets were created in a separate animation file, where each expression, phoneme, or feature pose was stored as a separate keyframe. Upon completing this file, a character artist would use our MEL Driven Key Generator (Figure 11) to set the driven keys automatically for each pose.

The Driven Key Generator worked by comparing the transformations of the keyframed pose to those of a default. When the script registered that a channel had changed from the default, it would set a driven key on that channel based on its changed value. The script relied on MEL's arithmetic functions to identify value changes, and its `setAttr` and `setDrivenKeyframe` commands to activate the drivers. Listing 2 shows some of the Driven Key Generator's sample code.

The drivers for our facial animations were stored on a model called the Control Box, shown in Figure 12. This hierarchy of cubes served as a visual outline of facial attributes and could also double as a second interface. For efficiency's sake, Ratchet, Clank, and all of our NPC characters had identical Control Boxes, though Ratchet's had many more active drivers.

We found our automated setup method to be advantageous for three reasons. First, it saved a setup artist from having to manually identify and key bones, channels, and drivers. Second, it assigned driven keys to changed channels only, leaving any non-affected channels free for animators to keyframe. Finally, it circumvented Maya's built-in driven key interface, which we found to be cumbersome and even unreliable when simultaneously assigning multiple bones and channels to a driver.

Regardless of method, facial animation played a vital role in breathing life into our gameplay characters. Again, MEL was instrumental both in granting our artists access to an advanced Maya feature and in optimizing our workflow. Whether a hero or an enemy, virtually every character personality in our game was strengthened through facial expressions. In turn, this enhanced interactions both with players as well as between the characters themselves.

End of Cycle

Like all character-driven projects, RATCHET & CLANK presented our animation team with a unique set of artistic and technical challenges. Our artistic philosophy was built on the understanding that our characters were the instruments through which a player would experience our universe. We knew that in meeting these challenges, our puppets would transcend mere game space and become the entities that our players would identify with, vilify, and even personify.

However, this philosophy needed to be coupled with practical methodology if it was to see our project to its conclusion. From this necessity grew our testing practices, MEL shortcuts, and real-time animation procedures. Throughout production, these methods removed many of the barriers that would otherwise have obstructed the artistic efforts of our animators.

As the Insomniac team cycles into our next project, we continue to refine and expand upon the systems and procedures we developed during RATCHET & CLANK. Though our procedures continue to evolve, our underlying goals remain unchanged. For in the end, we can only prove a technology's worth by an audience's response to our characters.

LISTING 2. Sample code from the Driven Key Generator

```
// The "if" gate checks for changed X-Translation values
// between the Default and Posed frames.

if ($txa != $txb)
{

   // Sets the Driver Attribute and the Current Joint's
   // X-Translation to their Default Values;
   // Sets a Driven Key Frame for the Default Values.

   setAttr $atnm $dr0;
   setAttr ($current + ".tx") $txa;
   setDrivenKeyframe -currentDriver $atnm -attribute
      translateX $current;

   // Sets the Driver Attribute and the Current Joint's
   // X-Translation to their Posed Values;
   // Sets a Driven Key Frame for the Posed Values.

   setAttr $atnm $dr1;
   setAttr ($current + ".tx") $txb;
   setDrivenKeyframe - currentDriver $atnm -attribute
      translateX $current;

   // Prints command summary to the Script Editor for
   // easy reference.

   print ($current + ": TX has been keyed for slider
      value 0: " + $txa + " and slider value 10: " +
      $txb);  print " \n";
}

// In this loop segment, $current is the current joint,
// and $atmn is the attribute name. $dr0 and $dr1 represent
// Default and Posed Driver values. $txa & $txb are the
// Default and Posed X-translation values, respectively.

// (Note: All flags are listed in their long forms.)
```

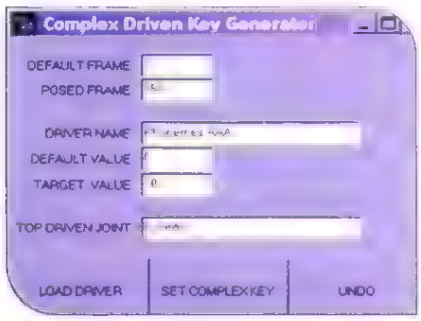


FIGURE 11 The Driven Ke... a preset facial pose, compared it to a ... pose, and assigned driven keys to the affected channels.

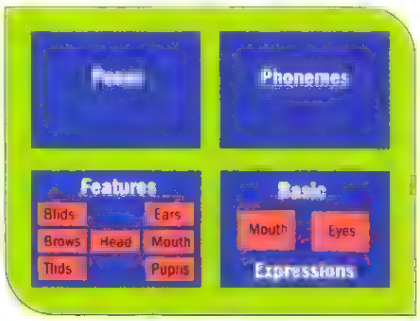


FIGURE 12 Each NPC ...

FOR MORE INFORMATION

BOOKS

Gary Fagin. *The Artist's Complete Guide to Facial Expression*. New York: Watson-Guptill Publications, 1990

Frank Thomas and Ollie Johnson. *The Illusion of Life*. New York: Hyperion, 1981.

RECOMMENDED MAYA TRAINING

MEL Fundamentals (formerly MEL for Artists): Information available at www.aliaswavefront.com, click on Maya Training under "Education"

Jennifer Olsen

Inside the 2003 Game Developers Conference

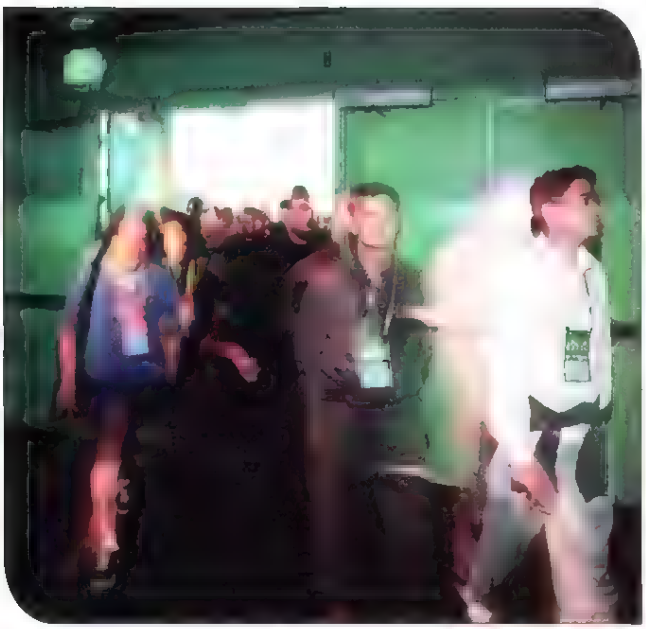
The 17th Game Developers Conference convenes again in San Jose from March 4 to 8 this year, and conference organizers find themselves in a quandary similar to that faced by many of their attendees: overcoming sequelitis. Like many developers working in today's sequel-driven marketplace, their challenge lies in satisfying returning customers who expect certain features and experiences that made forerunners popular, while innovating and refining enough to attract new customers and add value for those returning.

From content spanning the different conference tracks to the extracurricular parties and events, the organizers, CMP Media's Gama Network (which also publishes *Game Developer*), and the GDC Advisory Board have focused their efforts on honing GDC into an indispensable event for game developers. The events market is still feeling the effects of 9/11 and the economic slowdown, and many development studios that used to spring for annual trips to GDC, E3, and Siggraph are now scaling back their budgets to include just one or two events for employees. Others make the trip on out-of-pocket expenses, hoping that glittering pitch demo or polished résumé will pay off big returns. Some critics contend the event has become too "corporate," leaving the needs of small independents, and even the event's own grassroots beginnings, behind.

Taking into account the changing needs of a rapidly evolving industry, organizers are always looking at new opportunities to ratchet up the value of the event. The advisory board has made careful iterations in the content goals of the various conference tracks, while emerging markets are finding new prominence through programs such as the two-day GDC Mobile event.

Creating a stronger sense of developer identity is important, too. The International Game Developers Association (which functions as an independent nonprofit under a management contract with the Gama Network) develops its own conference track focused on developer and community issues, and the IGDA returns with the third annual Game Developers Choice Awards, a program that aims to create meaningful peer recognition for game developers. Meanwhile, the Independent Games Festival, now in its fifth year, continues to groom itself into a showcase for innovative indie game projects.

Aside from the din of the GDC Expo floor, the blur of parties, and the impromptu encounters in the hallways, the heart of GDC remains in the more than 300 lectures, panels, and roundtables presented, and in the speakers who variously swim or sweat through their Powerpoint slides for the betterment of industry knowledge. We caught up with a few of this year's speakers to find out what they have in store for attendees and what they as attendees think of this ever-evolving event.



GDC attendees descend on the Expo floor as the doors open

ADAM SCHNITZER
Senior Artist, LucasArts
Visual Arts Lecture: "How to Build a Better Cutscene"

Game Developer What will your session cover?

Adam Schnitzer I'll talk about the reasons for doing cutscenes, how to plan your cutscenes, the importance of previsualization, and the ins and outs of how to transition from gameplay to cutscene and back again. I will also spend a little time talking about production methods that can streamline the cutscene creation process. Because my background is as a layout artist at Pixar, my perspective is that of someone who is very concerned with cinematic design, so much of my talk will focus on that as well.

GD How do you see the role of cutscenes in games changing in the next few years?

AS There are so many different styles of games, it's hard to predict with certainty what the role of cutscenes will be. But for a certain style of story-based game, they are indispensable. With game engines getting more and more sophisticated and the consoles allowing us to create more filmlike environments, I can envision more immersive cutscenes and more seamless transitions in and out of cutscenes. With these higher-resolution in game environments that are evolving, we will eventually be able to dispense with prerendered cutscenes altogether, but I would guess that that day is still five to 10 years off.

GD The name of your session implies — fairly — that there is ample room for improvement in the quality of game cutscenes. Why do you think this area continues to vex developers?

AS It's important to be very clear on the purpose of the cutscene for your particular game. Cutscenes can do a lot to enhance gameplay, but gameplay always comes first. Having said that, I think there is a lot of ignorance of the fundamental principles of cinema in the game world. These are principles which were codified and haven't changed for the past 70 years or so. And when you go from gameplay to cutscene, you are stepping into the arena of cinematic storytelling. A greater awareness of cinematic structure and the power of the camera would help bring cutscenes to a higher level.

GD How many years have you been going to GDC?

AS This is going to be my first year at GDC.

STUART ROCH
Executive Producer, Shiny Entertainment
Production Lecture: "My Matrix Experience: A Survival Guide to Working with Movie Licenses"

GD What will your session cover?

Stuart Roch While working on the Matrix project, the film's interactive producer and I often talked about what a shame it would be if, once the project was completed, all the knowledge she and I had gained about the marriage of Hollywood and the gaming industry were to be lost. We actually felt a little bad, figuring that others in the business might be forced to learn through trial and error as we have had to these past couple years. So I hope to share all the lessons we have learned, so others in the industry can get a step up on a future licensed game. I'll cover everything from the initial deal all the way through the postproduction process.

GD What are the communication challenges developers face when working with a movie studio on a licensed property?

SR A number of communication problems can arise, from making sure the movie and game companies use the same terminology to describe issues, to coordinating schedules between two entertainment mediums whose pre- and postproduction schedules don't mesh very well. Perhaps the biggest communication problem that can arise is when a developer can communicate to the directors only by going through the studio office. This indirect communication channel can lead to delays and potential misinformation. In our case, the Wachowski brothers recognized this potential problem and set up communication channels directly between themselves and Shiny to make sure that accurate information and assets were flowing as efficiently as possible.

GD Do you feel that your experience on the Matrix project carries over well for future endeavors?

AS Not only has the development process been easier than we expected due to the unique communication channels, but we've all learned a lot about how Hollywood approaches similar development problems, and we applied some of their solutions to our development process. Hollywood is a mature industry, and through my session, I'll be sharing some of their techniques which may be of help to other developers.

GD What have been the most significant changes to GDC since you began attending?

AS The most significant changes I've seen at GDC are its growth internationally and the fact that we have so many respected overseas speakers flying over to share their knowledge and experiences. There's also been an incredible growth in new talent looking to work in this industry. GDC gives those people a chance to widen their perspective and really see what it's all about. They can meet the people that are making the games, they can meet some of the people they look up to, they can be inspired by the creativity of others, and they can take away concrete advice.

GD What's your favorite event at GDC? What's your least favorite thing about GDC?

AS I always really enjoy the lecture sessions. Unlike other industry shows throughout the year, this is the one stop I make where I always have useful takeaway from my industry peers. You never feel as though the time spent at GDC is time wasted. The thing that always seems to be a problem with GDC is that it inevitably coincides with some sort of crunch time on one of our projects.

Cover every angle.

Secure your goal.

Your goal is to protect your PC games from the mounting threat of piracy to maximize your sales window

New **SecuROM Trigger Functions** significantly increase the odds in your favor by allowing you to put multiple lines of defense in your software title, making each protection code unique and rendering Internet piracy and reverse-engineering attempts practically useless

And our most recently launched **Secunda authentication signature** gives you an edge no other system can match: added protection against digital cloning

SecuROM offers the highest possible CD-ROM copy control security, along with industry-leading drive compatibility and unique ease-of-use for implementation

For more information, call **831-475-7801**, send e-mail to **dmark@disc.sony.com**, or visit our Web site at **www.securom.com**

A Technology by **Sony DADC**

securom

KATIE SALEN
Independent Game Designer
ERIC ZIMMERMAN
CEO, gameLab
Game Design Lecture: "Breaking the Rules of the Game"

GD What will your session cover?

Katie Salen We are presenting a lecture on rule-breaking and its relevance for game design. Like it or not, rule twisting, bending, and breaking are part of games — but rule-breaking can be positive or negative, and game designers have a lot to learn from it. We will look at the formal, social, and cultural ways that players break rules and how rule breaking can be integrated into the game design process.

GD Why do players want to break the rules?

KS Players break rules for all kinds of reasons, but they generally do so not to break or end the game. Instead, most creative forms of rule-breaking introduce excitement, variety, strategy, and fun into the game.

GD What can game designers learn from rule-breaking?

KS Because games are systems, it's always possible for players to drive a wedge into it, bending the system into a new shape by breaking the rules. We challenge game designers to find ways of including this natural desire of players into their game designs. When given the right tools, players will transgress the rules of the game in pursuit of alternate forms of expression. How can this desire be enhanced or slowed, modified or transformed, by the design of the game itself? This is what we will explore in our talk.

GD How many years have you been going to GDC? What has been the most significant change since you began attending?

Eric Zimmerman My first GDC was 1998. It was the year in Long Beach

when the entire conference played a giant game of dart-gun ASSASSIN. The conference has lost that kind of wonderful folksiness, but that is inevitable as the industry grows. For the last two years at the GDC, gameLab has tried to bring back a bit of that spirit with our own massively multiplayer off-line games (BITE ME in 2001 and LEVIATHAN in 2002). Look for our new game this year in the IGDA booth.

GD What's your favorite event at GDC?

EZ The best moments at a conference always happen in the interstices between the organized talks, meetings, and parties. Surprise encounters between sessions, gab sessions at cheesy hotel bars, and late-night hotel-room game design debates are what make GDC so fantastic for me. I just wish there was more time.

JAKE SIMPSON
Psychology Programmer, Maxis
Programming Lecture: "Animation System Implementation: What Works and What Doesn't"

GD What will your session cover?

Jake Simpson My session involves discussing approaches to building state-of-the-art animation systems, heavy on the practicalities of what works and what doesn't, what's worth experimenting with and what's not. This information is drawn from experience in working with animation systems for third-person games (HERETIC II), and creating new animation systems for networked first-person games (Ghoul 2 for SOLDIER OF FORTUNE II and JEDI KNIGHT II) and for next-generation SIMS products.

GD What's the single biggest takeaway you got from developing Raven's Ghoul system that you think can save developers time in planning animation system features?

JS "Give the animators and coders more time to learn how to use the new technology." Game development being what it is, it usually takes two or three games before new technology matures enough for content creators to really know where the limits are and how to use it all in the most efficient manner. The earlier you get the tech in the hands of the content creators, the better look

ing and more spectacular the results will be. Great tech depends on artists and content creators building stuff that uses it to the best of its ability — you need everyone to be on the same page for it all to come together.

GD How many years have you been going to GDC? What has been the most significant change since you began attending?

JS I've been going for about four years now. From where I sit, the most significant thing is what the GDC Advisory Board is doing. They've made such an effort this year to try to be specific in what they present, in terms of actually getting value out of the lectures. I figure if I have to take notes at a lecture, then it's worth my time, and they seem to be going all out for this effect this year.

Also, every year I feel like we are starting to get more attendees from the rest of the world, notably the Japanese. We all have something to learn from people who have such mastery of our craft, and I applaud this international flavor.

GD What's your favorite event at GDC? What's your least favorite thing about GDC?

JS My favorite events are the informal get-togethers that happen after lectures and roundtables. Being able to pigeonhole people with proven experience about a specific issue you may be having, or even just bending their ear about something that interests you, is great. It saves time, wasted research effort, and usually nets a suggestion you would never have had yourself.

Least favorite? Well, let's be honest, it's the hangover each morning. What's a GDC without some company-sponsored parties?

DAN SCHERLIS
CEO, Etherplay
Business and Legal Lecture: "Doing Business with the Telecom Industry: Understanding Their Deal Terms, Culture, Rites, and Rituals"

GD What will your session cover?

Dan Scherlis Many developers are fascinated with mobile games, for many good reasons. And many developers are having trouble getting attention from telecom types, not to mention getting deals and getting paid. I am not being glib in the session title; I do feel that the major issues are downright anthropological. The two industries speak and act differently.

GD What has the telecom industry learned about game developers in the past year, and has it altered their approach to these new content creators at all?

DS Telecom has been a motivated and attentive student of games and of game developers. In the last year I have heard far less talk of games as a commodity and more respect for the value of quality product. This is bringing about better deal terms.

That said, we are still of different worlds, and there is so little history of deal-making between us that we do not have much precedent for basic terms and structures of our deals.

GD What's the single biggest gotcha that awaits game developers making their first foray into dealings with telecom companies?

DS Communication — especially style of communication: Game developers use whiteboards; telecom types use Powerpoint

Also, a developer should not assume that a telecom executive knows anything about games. If you offer an RTS with FPS action but RPG depth, with downloadable mods and ortho view, you will not be understood. If you give examples like WARCRAFT, HALF-LIFE, and FINAL FANTASY, you will continue to strike out. Test your pitch on a smart nongamer. Your parent or spouse will have too much of a clue; try your dentist or accountant.

GD **There have been many losses, and much pain, amongst mobile-game startups. Where do you see an opportunity for profit from mobile game development?**

DS Mobile games are echoing the history of Internet games. We see many basic mobile games, which are hard to differentiate. Over the Internet, we saw many ad-supported games, few of which earned good money for their developers. The games that make big money online are fundamentally different: they are of premium quality, their designs exploit the network and include persistent social structures, and they have subscription economics. I believe that this description will apply to the most profitable mobile games. This is why several online-game veterans are being attracted to mobile platforms; we see an opportunity to build profitable communities, without the multi-million-dollar, multi-year development cycles.

The greatest challenge is to the designer. We need to design explicitly for this new medium, rather than trying to sell either shovelware or obvious but derivative and shallow distractions.

GD **How many years have you been going to GDC? What has been the most significant change since you began attending?**

DS I started attending GDC in 1992, I think, when I joined Papyrus. I've missed only one or two since then.

The most dramatic change has been the astounding growth. As the common complaint goes, this growth threatens the sense of community and collegiality that attracts many long-time attendees.

I can gripe with the best of them, but I have been pleased by the degree to which the community survives. The San Jose Convention Center has been a challenge, many industry veterans stay home, and the sheer size is daunting, but I am always delighted by the many chance encounters with colleagues at GDC.

TOMMY TALLARICO
President, Tommy Tallarico Studios
Audio Panel: "Orchestral Panel"

GD **Who's on the panel and what do you hope to cover?**

Tommy Tallarico We will have some of the best orchestral composers for the videogame industry on the panel: Jack Wall (MYST III), Clint Bajakian (of LucasArts fame), Jeremy Soule (HARRY POTTER), Bill Brown (RAINBOW SIX), and Dan Irish (producer of MYST III). We may also have a few special guests and surprises!

Last year's orchestra panel was really tailored toward composers. This year we felt it very important that the producers and designers know about live orchestral sessions as well, so we are tailoring it more for them. Budgets, production dos and don'ts, how to convince your boss, the real value of using live players, union vs. buyout, as well as other things, will all be covered in this panel. It is important for the project decision makers, milestone schedulers, and purse-string holders to find out how to go about getting live music in their projects.

GD **How do you think producers and audio professionals should best go about deciding when to use orchestral performances in games and when not to?**

TT Style of music and game genre is one big contributing factor. Not every game warrants live orchestra, but most would benefit greatly because of it. Publishers are looking to differentiate themselves to the consumer from a quality standpoint, and using a live orchestra can help do that. Live orchestras don't cost hundreds of thousands of dollars to produce, either — there are many different alternatives. You can get a 50-piece live orchestra recorded for less than $1,000 per minute of music.

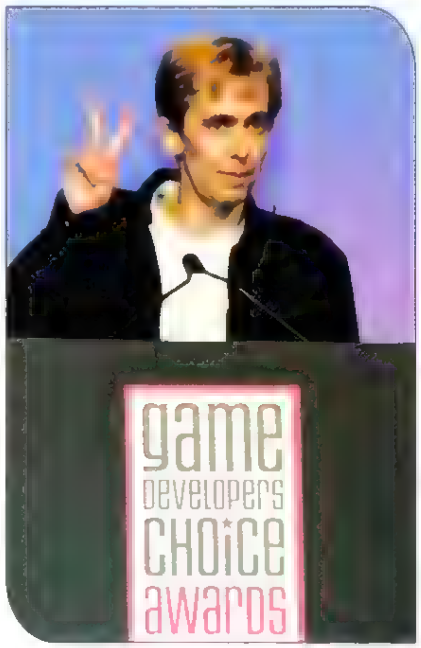
GD **How many years have you been going to GDC? What has been the most significant change since you began attending?**

TT I've been going since 1993, when I used to sneak in and just go to the parties. Then I got smart and became a speaker. The biggest change besides the obvious sheer number of people would be the emergence of a huge audio community coming out to GDC. In the early years you could count the number of audio people on three hands. Since the Audio Pass was created, hundreds of interested audio professionals and nonprofessionals have come seeking knowledge and understanding of this crazy profession. In audio you have to deal with three important elements in production: creative, technical, and business. GDC is a place you can go to learn about all three.

GD **What's your favorite event at GDC? What's your least favorite thing about GDC?**

TT My absolute least favorite thing about GDC is trying to get a room every year. My favorite event is the Game Developers Choice Awards ceremony. It's really nice to see the developers get the recognition they deserve from their peers.

For the most up-to-date GDC 2003 information visit .



Lionhead's Richard Evans accepts last year's Game Developers Choice Award from the GDA for Excellence in Programming for his work on Black & White's AI

THE 3RD ANNUAL GAME DEVELOPERS CHOICE AWARDS

Nominations open January 2nd, 2003

"My passion is games, and having my life's work recognized by developers worldwide is the most meaningful honor I could receive."

– Yuji Naka, Sega
Lifetime Achievement Award recipient

game developers choice awards

Started in 2000 with the idea that there is no greater honor than to be recognized by one's peers, the Game Developers Choice Awards are game development's most prized honors.

All professional game developers are eligible to nominate the developers who transcended the state of the art in 2002 at

www.igda.org/awards

Sponsored by:

RenderWare

Giving Life to Ratchet & Clank

Enabling Complex Character Animations by Streamlining Processes

At first, we were thrilled. As character animators, we couldn't have asked for a better project. There were two heroes, dozens of enemies, scores of NPCs, and more than 100 character-driven cutscenes. Enthusiasm and artistic latitude made it all ours for the taking.

But staying true to our shared vision of Ratchet & Clank meant that our digital actors needed to become more than mere cycling automatons. We regarded each character as an intermediary through which we could reach out to players and draw them deeper into our universe. This meant our characters needed to blend physically into their environments, emotionally into their situations, and expressively into our narrative. It was on these principles that we based both our objectives and our standard of success.

Our team acknowledged that a rift existed between the level of complexity we desired and the time we had scheduled to implement it. In order to surmount this obstacle, we developed several methods for using Maya, our artistic skills, and our time more effectively.

This article will discuss these methods both in terms of their functionality and their implementation. To this end, it will provide technical details on our testing practices, our MEL shortcuts, and our real-time animation procedures. Furthermore, it will explain how each of these methods saved us valuable production time, enabling us to achieve our artistic goals.

Testing with Prototypes: Why and How

Part of achieving our goal of tying our characters closely to their environments and gameplay meant prototyping low-resolution versions of our characters and their respective animations. Like coalmine canaries, we sent proto-models into our new levels to nose out potential animation, programming, and design problems. We relied on prototyping throughout the course of our production as a means of refining a character's move set. This process of refinement was key to winnowing down unworkable ideas before animating a character's high-resolution incarnation.

As a rule, our prototypes emphasized function over style. And although we set the aesthetic threshold low, these previsualization models still needed to be built and animated accurately enough to function as valid test cases. For the animators, this meant that prototype characters needed to jump to their correct heights, attack to their design specifications, and run at their proper speeds.

Generally, we created prototypes using a character design sketch as a guide. These proto-characters were constructed with primitive objects and only roughly resembled their future incarnations, as you can see in Figure 1 (on page 32). Since previsualization models were so simple to construct, every animator could assist in building them, regardless of their modeling experience. Accuracy was required only in the representation of the character's height, proportions, and posture.

For the most part, our prototypes had extremely simple skeletons: all geometric components were assigned to a single bone with no special deformation. Though such simplicity made for blocky-looking models, in practice our animators had all the flexibility they needed to test out a move set.

Animating our proto-characters was similar to sketching a traditional pencil test. Although animators were given a designer-approved move set, it was understood that animations needed only to be rendered into their roughest forms. One pass was often sufficient, as polish and overlap were unnecessary.

The areas where precision did count were timing, measurement, and interaction with other characters. As they have the greatest direct impact on gameplay, these attributes were considered critical to testing a new character's behavior accurately.

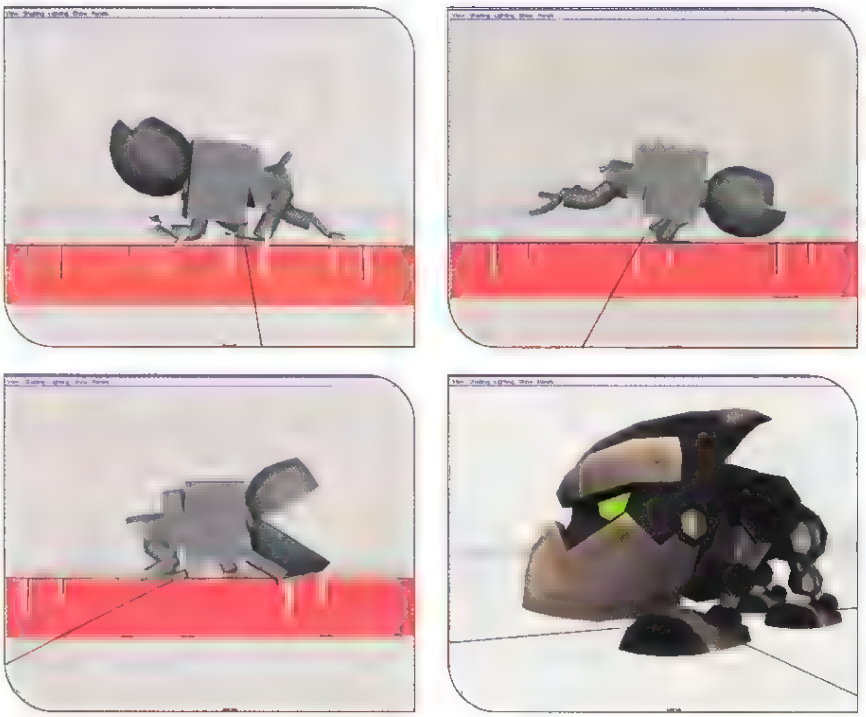
Timing has a major effect on both the readability of an animation and on gameplay. From a distance, a poorly timed idle can look muddy. An attack animation can be too slow to make an enemy a worthy opponent, or too fast to be registered. Emphasis or a lack thereof on just a few frames can make or break any animation, especially within the short cycles of the real-time universe we were creating. We discovered that by testing and fine-tuning our timings in the prototype stage, we could often avoid reworking polished animations on final characters.

Making sure that proto-characters adhered to design measurements was also important. For example, if the design document called for an enemy to attack at a range of 4 meters, animators would ensure that the prototype did exactly this. Designers could then get an accurate idea of whether an enemy traveled at the correct speed, was tuned to the appropriate difficulty, and was scaled appropriately in relation to the main characters.

Prototyping also gave us a means of pretesting character behaviors and interactions. Whether it was with Ratchet or Clank, with the environment, or with another character, proto-models provided invaluable early glances at interactive behavior. For artists, programmers, and designers, previsualization served to telegraph character behaviors both in terms of their technical feasibility and their gameplay value.

Ultimately we found that our previsualization process was beneficial not just to animators but to our design and programming staff as well. It gave our programmers a head start on coding gameplay, while designers could test, tune, and ask for changes at a very early stage, allowing room for refinements.

Prototyping saved animators time and energy that otherwise would have been spent painstakingly modifying or redoing final multi-pass animations. It provided a relatively simple means for evaluating character behaviors with respect to their timing, specifications, and interactivity. Moreover, it provided our animators with a practice run, complete with feedback, before moving on to a high-resolution character (Figure 2).



FIGURES 1A–C The Dog Charger prototype [illegible] used to prote[illegible] including its walk, run, and attack. FIGURE 2 (bottom left). Th[illegible]

MEL Shortcuts: Automating Our Setups

Maya Embedded Language (MEL) scripts were essential for bridging the gap between the level of complexity we desired and the time we had scheduled to implement it. Through MEL scripts, we were able to streamline setup operations, customize animation processes, and level our technological playing field.

Two such scripts (examined later in this article) allowed our team to take advantage of driven key functionality that otherwise would have been too cumbersome to animate or too tedious to rig by hand. Another tool enabled our artists, regardless of technical experience, to fit characters with IK systems automatically.

Most of our bipedal characters had leg setups like the one pictured in Figure 3. As seen in the hierarchy (Figure 4) our legs had standard hip, knee, and ankle joints, a heel joint, and two to three bones in the feet. (For clarity purposes, please note that we referred to our foot bones as "toes.")

Our IK-rig consisted of three to four RP (Rotate Plane) IK-handles. These connected hip-to-ankle, ankle-to-toe, toe-to-toe and/or toe-to-null. All were configured into a hierarchy (Figure 5) that specified relationships between the IK-handles, a set of locators, and several NURBS constraint objects.

Though relatively simple, setting this IK-system up by hand for every NPC, enemy, and prototype would have taken more time than we had. Moreover, we knew that this time would be better spent bringing our characters to life.

An actual tools programmer might scoff at the artist-authored MEL script we developed to make our leg chains. In the end, however, our "IK Setup Tool" reduced an hourlong technical chore to a simple task that took seconds. Furthermore, the script did not require setup expertise, and our relatively simple code could be customized and refined entirely from within the art department.

Using the IK Setup Tool (Figure 6) was a three-step process. First, an artist checked their characters' leg joint names against the tool's presets, making any necessary changes. Next, a scale factor for the constraint objects was entered,

Developing Sequels: The Designer's Dilemma

Ensemble Studios' AGE OF MYTHOLOGY

generously as "challenging."

first game.

IAN FISCHER *Ian joined Ensemble Studios as a designer in 1997, just in time for the final stages of* AGE OF EMPIRES. *In his spare time he likes playing games and arguing. He can be reached at ifischer@ensemblestudios.com.*

GREG STREET *Greg was a marine biologist up until he discovered* AGE OF EMPIRES *in 1998 and ended up joining Ensemble Studios to work as a game designer. Feel free to send your marine crustacean questions to gstreet@ensemblestudios.com.*

GAME DATA

PUBLISHER: Microsoft

NUMBER OF FULL-TIME DEVELOPERS: 50 total employees: 15 programmers

CONTRACTORS: 10 quality-assurance contractors, no contract programmers

LENGTH OF DEVELOPMENT: 30 months

RELEASE DATE: October 31, 2002

TARGET PLATFORM: PC

DEVELOPMENT HARDWARE: Ranged from 300 MHz Pentium IIs with 64MB RAM and TNT1 graphics cards to 1.7GHz Pentium 4s with 2GB RAM and GeForce 4 graphics cards

DEVELOPMENT SOFTWARE: MS Visual Studio 6, Source Safe, 3DS MAX 4.0, Photoshop

NOTABLE TECHNOLOGIES: Granny, Bink

PROJECT SIZE: 1,500,000 lines of code

the series, plus a few expansions, and are starting to grumble for something different. At the same time, removal or alteration of any existing feature will be met with ranting e-mails, forum petitions, and overturned cars in your parking lot, so this is also the time when preserving everything the old games had becomes vital. The engine and tools you developed for the first game and advanced in the second are behind the technical curve by this time, so now you need to add developing and learning new ones to the to-do list. And, at some point, you're going to get a visit from a graduate of the this-trend-will-continue-forever school of projection who, armed with charts showing that title One moved a million copies and Two moved 3 million, will tell you Three should move 9 million copies.

This is where the design team was at the start of AGE OF MYTHOLOGY. The big question that haunted us was: Wow, what are we going to do to top AGE OF KINGS?

Ensemble Studios' projects are ambitious, and we ratchet up the ambition with each new project. As the scope of the project grew, the size of the team grew. We were developing a new engine and a new multiplayer online service at the same time that we were developing a new game, a game in which we wanted to incorporate new features, such as God Powers and myth units, and a more ambitious single-player campaign.

Rather than restate the all-too-common problems of having more ambition than resources, of having marketing push for content before it's ready, and of having personnel problems every company goes through from time to time, we find it more useful to focus on design aspects in this article. Designers are Ensemble's vision-bearers, but we don't get to just ram our ideas down everyone else's throats (as attractive as that power sounds at times). The designers must keep the project in scope, keep the artists and programmers from killing each other, and make sure feedback is heard without devolving the game into a design-by-commit tee project.

What Went Right

1. Iteration. Ensemble's basic design process is to get the game playable early and then tweak it until it's fun. This applied to virtually every feature in the game. Some features changed a million times, and we were willing to abandon things that just didn't work, even when it was painful.

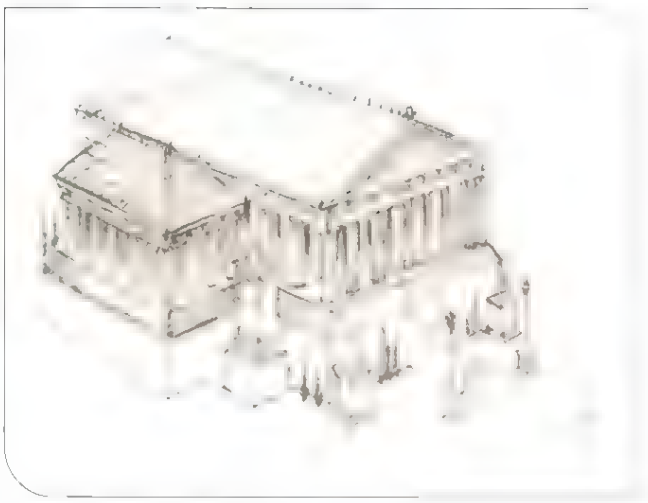
AGE OF MYTHOLOGY's God Power feature is a good example of this process in action. On paper, our initial concept of God Powers and Heroes sounded good: Heroes would have buttons on the interface to target God Powers wherever the selected Hero happened to be — simple enough.

Unfortunately, when we got the feature in the game and started playing with it, it was awful. Having to have a Hero in the place where you wanted a God Power devolved all combat tactics to selecting all your units and clicking on the enemy Hero. This led to Heroes constantly getting killed, prompting comments like, "The Heroes don't feel heroic." Additionally, with all God Powers targeted with your Hero, if you called down a meteor, it would land on his hand. It didn't damage him, but it didn't look good.

We tried a new model where the Heroes built "lightning rods" for the God Powers (so players could kill something other than enemy Heroes to stop an opponent's God Power, and so that Heroes could get out of the way of their own God Powers). This wasn't fun. We tried another model where you could buy all of the God Powers with resources, like most everything else in the game. This wasn't fun. We tried a dozen more models and variants.

Finally, after a lot of trial and error, we hit on the model we shipped with. Heroes were divorced from God Powers and made to be the thing used to kill myth units (which feels decidedly heroic). God Powers were moved to the main interface, and we made them single-use only, which made them feel large and important and kept them from landing atop Heroes at every use.

Units were first conceptualized to make sure that they both fit with the common art style, and also could be easily discerned from similar ones in the game



Ensemble Studios worked hard to capture the building detail that was a hallmark of AGE OF EMPIRES in the new 3D engine

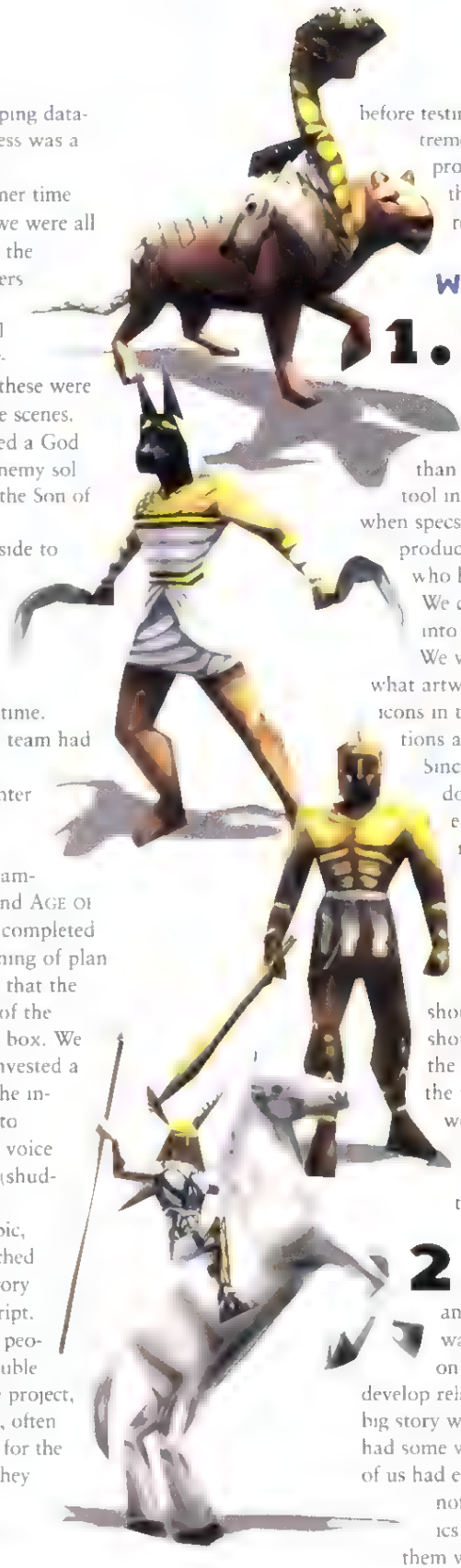
Because God Powers were so important to the vision of the game, we couldn't just yank them from the title after the sixth or seventh different model didn't work. Instead, we just continued to try different systems, brainstorming and then implementing models. Because a new approach often required new code and new art before it could be evaluated, we ended up throwing a lot of work away to achieve our end result. But we did achieve an end result we're very happy with. Ours is emphatically not an efficient process, but it continues to work for us.

2. Everyone play-tests. It's amazing how many developers rely on outside testers to tell them if the game is fun or not. Outside feedback is vital in the later stages of a project, but if your entire game is designed by polling the fans or beta testers, you end up with a mushy game with no vision. At Ensemble, everyone play-tests the game at least once a week. This strategy keeps the team bought in to the game that's being developed. There are mandatory, assigned play-test times in the morning and multiple pickup games in the afternoons or evening.

We have found that these play-tests are instrumental in keeping the team informed on the state of the project, giving them ownership of the process, making sure bugs don't slip through the cracks, and figuring out when the gameplay is fun enough to ship. The first implementation of God Powers described in the previous section made sense until it got out in front of our coworkers, at which time a mob brandishing torches and pitchforks strolled into our office. If the designers had relied solely on our own instinct about the model, we likely would have shipped with it.

3. Small meetings. For AGE OF EMPIRES and to a lesser extent AGE OF KINGS, we kept the entire team involved in the high-level design. In one particularly long meeting for AGE OF KINGS, we tried, as a company, to come up with a design for herd animals. Past a certain number of attendees, it became unmanageable to go around the room even once. So, as these meetings got longer, we tried to keep focus by including the various department leads and trusting them to relay the feedback of everyone on their respective teams. However, on a project the size of AGE OF MYTHOLOGY, even the leads' meetings could have a dozen attendees, making it harder to reach a consensus on any of the issues.

Eventually we refocused the leads' meetings on task management and progress reports and implemented a new series of meetings for design brainstorming with a core group of only four to five people, half of them designers. Features, such as the list of civilization bonuses, myth unit abilities, and God Powers, were all compiled in these meetings. When necessary, we took these meetings offsite to make sure we could get our business done without distractions. We found that e-mail wasn't efficient enough, and as busy as everyone was, impromptu meetings weren't always possible. We had to be formal about scheduling these conferences, which we ended up arbitrarily calling "small pets" (after "pet features," since we needed a name that didn't sound like we were excluding anyone). Because it was important to our process that everyone have a chance to give feedback, we would announce the decisions made in "small pets" meetings to the company at large. We heard people's concerns and ideas, incorporated any changes we thought necessary, and then implemented the design into the game. Everyone still had a voice, but ultimately we couldn't rely on a large group to come up with design implementations as fast as we needed them.



4. Data-driven tools. Developing data-driven tools early in the process was a strategy that really paid off for AGE OF MYTHOLOGY. It took a lot of programmer time away at the start of the project, when we were all anxious to begin playing the game, but the resource hit paid for itself when designers could implement new content without having to wake up the programmers all the time. For example, although we targeted 36 unique God Powers, some of these were implemented in similar ways behind the scenes. Once the programmers had implemented a God Power to switch units, we could turn enemy soldiers into pigs, or allied Pharaohs into the Son of Osiris, all through data.

There is, however, a potential downside to spending so much effort on tools. For one, instead of working on higher-visibility game features, programmers spend their time on tools that players may not see. Second, you might be trading programmer time for designer time. Near the end of the project, the design team had all the tools they needed to implement some features but lacked the time to enter the changes.

5. Focus on campaign. The campaigns for AGE OF EMPIRES and AGE OF KINGS were fun but lackluster, largely completed by one or two designers. At the beginning of planning AGE OF MYTHOLOGY, we decided that the single-player campaign would be one of the game's big features on the back of the box. We hired several new content designers, invested a lot of time in custom animations for the in-game cinematics, and made two trips to Hollywood to work with professional voice actors instead of using local talent or (shudder) our own overacting.

We had never before attempted an epic, character-driven script, and we approached the task in epic fashion, appointing a story committee to review progress on the script. (In retrospect, trying to please so many people so early in the project was more trouble than it was worth.) Near the end of the project, the designers working on the campaign, often with the artists working on animations for the cinematics, met several times a day so they could all keep in touch on progress. The lead designer documented everything, ensuring changes were made before testing the various scenarios again. It was a tremendous amount of work at the end of the project, when we could scarcely afford it. But the work paid off, and we delivered a well-received single-player campaign.

What Went Wrong

1. Design drove too much. Sure, the design department had all of these fancy tools, but in the end, designers ended up doing a lot of the work that a programmer might have been able to do faster than it took the programmer to develop the tool in the first place. We had early frustrations when specs didn't pan out as intended in the end product, coupled with the arrival of new people who had not worked with us on a title before. We compensated by heaping so much detail into specs that they were often not even read. We went so far as to provide descriptions for what artwork should be associated with the various icons in the scenario editor, and the various locations and states for those buttons.

Since all they were doing was connecting the dots on someone else's feature, new employees did not feel empowered. As a result of days spent writing things such as, "When you click this button, it should appear depressed until the user releases the mouse button, at which time it should revert to looking un-depressed; clicking the button in this manner should cause a sound to occur, the sound should be kind of like a twig snapping...," the design department took up drinking in the middle of the afternoon. In the future, we plan to keep design driving the process, but at a higher level. We will trust people at the implementation level to fill in the details.

2. Scriptwriting n00bs. We knew we wanted a script that had all the scope and drama of *The Iliad*, and we knew we wanted characters who could slap each other on the back, make fun of each other, and develop relationships over time. In short, we needed a big story with a lot of dialogue. While the designers had some writing experience in various forms, none of us had ever tackled a script like this. We also had not yet figured out how the in-game cinematics would work, or how long we could make them without boring everyone.

Rewarding Innovation in Independent Games

See the 5th Annual Independent Games Festival Competition finalists at www.igf.com

Quick Facts About the 2003 IGF Competition:

Entries:	73
Average development cost:	$36,770
Average development time:	14 months
Number of games with $50,000+ budgets:	10
Number of games with $0 budgets:	25
Number of games in development for less than 1 month:	3
Number of games in development for more than 2 years:	9
Percentage of games developed by teams with high hopes for fame and fortune:	100%

The 5th Annual Independent Games Festival
March 6-8, 2003 · San Jose, California

Craft sponsor, Innovation in Game Design
Microsoft DIRECTX

Craft sponsor, Technical Excellence
intel

Josh and Dan Goldstein, members of Shizmoo, accepting the Audience Award for KUNG FU CHESS at the 4th Annual IGF.

INDEPENDENT GAMES FESTIVAL

GamaNetwork

High-polygon opening cinematics are demanded by the fans. This time around, Ensemble experimented with working with an outside contractor. Those experiences might make for an entire Postmortem by themselves.

We had no idea what we were doing. We did it anyway. The result was a lot of revisions to satisfy different opinions about how a story or characters should work. Everyone had their favorite bit character or script fragment that was impossible to delete, and deleting anything threatened to collapse the intricate story line. We eventually had to take a step back and revise with a much smaller group of just two people. In the future we will keep early feedback to a smaller group, which we hope will get us closer to nailing the correct length, number of characters, and plot intricacies with fewer revisions.

Developing this sort of material at both a high level of quality and in something resembling an efficient manner demands some pain and experience that can only be gained by actually doing it. If you're planning on doing this sort of project and haven't done it before, double your estimates for everything related to the project, then halve your plans for the content (number of characters, lines of dialogue, number of scenarios, number of special art objects, and so on).

3. Consensus is hard with large groups. Consensus is the basis of the game design process at Ensemble. This company philosophy worked exceedingly well when there were only a dozen of us. Even when we grew to 20 or so people, getting everyone in a room (we usually all had lunch together) and hashing things out worked well. As the size of our team grew, however, it became increasingly less efficient to get everyone in a room several times a week. Even worse, we stalemated a lot more and started to resort to compromises to placate all involved, so some of our design decisions began to result in bland design-by-committee game models ("oatmeal design," in our parlance).

The models used for the in-game cinematics needed to match their low-polygon equivalents that were seen during gameplay.

We settled on a strategy where the design department would gather everyone's feedback, mull it over, and then make the call. It was difficult changing our mode of communication in the middle of the project, and some of the squeakier wheels protested that there was a design black hole that swallowed up their feedback. This prompted us to redouble our efforts at documenting changes and communicating them to the team, but this was an ever-widening gyre; for every e-mail you sent out explaining a decision, you got five replies that disagreed with various points of your logic and required a response. Eventually, we got to the point where we had to make a decision: the design team could either do the work we needed to do to and complete the game, or we could explain and defend every decision we made.

We came to think of our larger team size as simply a variable — it changes how we go about being consensus-based, not whether or not we use a consensus-based approach. Either way, we were committed to our consensus-based process. Our plan at this stage was to formalize what eventually worked during the latter portions of AGE OF MYTHOLOGY with our "small pets" meetings. Our new projects are now built around small, nimble groups with representation from all of the disciplines.

UNITING THE GLOBAL GAME DEVELOPMENT COMMUNITY

SURVIVING CHANGE IS OUR TOUGHEST CHALLENGE...

The huge scale of today's development projects is forcing game developers to change quickly from a small, loosely organized group of individual performers to an efficient production group — while retaining creativity. Surviving that change is the toughest challenge developer groups face now.

Many game developers have already achieved this. Those who don't will not survive.

The IGDA's Business Committee hopes to raise the level of professionalism and efficiency in the development community at large. As an IGDA director and co-chair of the Business Committee, I'm working on projects that specifically help developers overcome that hurdle.

More professional, more efficient developers will mean fewer conflicts with publishers, more alignment on creative vision, and cost efficiencies, which can translate into either less expensive game development or bigger and better games.

KATHY SCHOBACK

KATHY SCHOBACK
DIRECTOR OF EXTERNAL DEVELOPMENT & PUBLISHING
SEGA

Game developers must play a major part in helping the industry evolve.
Join the IGDA and get involved with the committees!

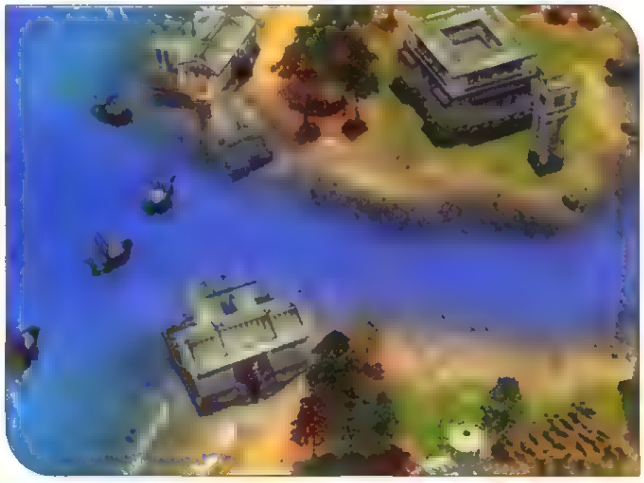
www.igda.org/join

The IGDA is sponsored in part by:

Microsoft Research Alias wavefront

igda international game developers association

One of the goals of AGE OF MYTHOLOGY was to design a beautiful living world



Ensemble uses bright colors to make inviting landscapes they hope players will want to spend time in

These groups have the ultimate decision-making authority but also the responsibility for gathering feedback from the entire team, explaining and defending decisions, and building a general consensus.

4. How different is "different"? The well-known, inherent risk of sequels is that you need to keep what is popular about earlier products while still offering something new to justify the purchase of a new product. The AGE OF EMPIRES games are large and complex, and we knew we couldn't take AGE OF KINGS and layer several new game features on top of it; we had to pull out some systems and change things to make a game that was "different." While some of us were (or thought we were) clear on what "different" meant, there were many other definitions floating about. For some members of the team, different meant, "AGE OF KINGS, but the knights look like Minotaurs." For other team members, different meant, "There are no units, and you control the game with your mind." When a new feature didn't work right away, the differences of opinion led to a lot of pressure to revert to the tried-and-true.

For example, we went through several variations of our population model. Earlier implementations lacked houses, which for some of the team was just too great a departure. The team quickly split into two camps, one that argued for even more change in the gameplay, and another that was scared that we were moving too far from the game our fans loved and expected. The fans are not particularly forgiving in this respect, and once the game was out, they applauded some of the new features while protesting about even the smallest features that were removed, such as choosing your player color.

Ultimately, there is no ideal solution to the problem of defining what is different; games today are too complex to be fully defined by a vision statement, so there will always be some degree of opinion to factor in. We plan to try to mitigate these disagreements in the future by attempting to answer the big questions such as "How different?" early on, and then keeping these guidelines in front of the team for the duration of the project.

5. Unfinished tools. Because we were dealing with a new engine and there was no shared code from our previous titles, we made the right decision to reserve a lot of time early on to develop tools. Unfortunately, since the tools were for internal use only, it was easy to move people off of those tasks when other problems came up. Several tools were never completely finished, and the customers for those tools (typically designers) waited up until the final days of the development cycle to see if improvements of incomplete features could be added. While waiting for the tool, we hacked a lot of content in place, figuring that it was better to do some work than sit idle. A good example was the AI for computer opponents in scenarios, which came online very late, after the designers had hacked in fake AI using scenario triggers. This kind of work was difficult to unhack and left us with less time to iterate than we would have liked.

For the future, we (like everyone else in the game business) need to remember the value of tools and not skimp on their development time. We additionally need to pay particular attention to those tools that might ship with the game, such as the scenario editor, which was never completed to our satisfaction.

Here to Four

The good and bad of getting there aside, the end product of all this is our Three, AGE OF MYTHOLOGY. Everyone at Ensemble Studios is immensely proud to have contributed to this game, and we're now looking forward to the opportunity to figure out what type of beast Four will be.

At this early stage, we only have one question on the board: Wow, what are we going to do to top AGE OF MYTHOLOGY?

8 years of GameDeveloper for just $189

Each CD-ROM contains:

- ✓ Full text of feature articles, columns, and product reviews
- ✓ Articles, artwork, and diagrams exactly as they were printed
- ✓ All code and applications that accompanied the articles

Great content. Great prices!

NEW

✓ May 2001 to April 2002	*$49*
✓ May 2000 to April 2001	*$39*
✓ May 1999 to April 2000	*$39*
✓ 1994 to April 1999	*$99*

Or get everything:

✓ Premier issue to April 2002	*$189*

Order by phone:
800.444.4881 (US) or 785.841.1631 (international)

Order online:
www.gamasutra.com/store

Sponsored by: Maya™ Alias|wavefront

GamaNetwork

SOFTWARE DEVELOPMENT CONFERENCE & EXPO

SD WEST 2003

SANTA CLARA CONVENTION CENTER

MARCH 24-28, 2003

BUILD BETTER SOFTWARE

JAVA, C++, .NET, Web Services, XML, Process, & Management

Learn the [illegible] knowledge and [illegible] techniques [illegible] apply to your job

Explore where the industry is headed during visionary keynotes, panels and BOFs

Network with 50 renowned faculty, exhibitors and peers at [illegible] and mingle at activities such as Awards, Developer Bowl, and parties

SD West 2003 — Five full days of [illegible] 150 classes, [illegible] and networking events

Join thousands of developers and more consistently chosen SD as the one conference they attend for over the past sixteen years

Register [illegible] www.sdexpo.com

Early Bird Discounts!

SAVE up to [illegible] by January 17th!

SAVE up to [illegible] by February [illegible]!

Featuring

WEB SERVICES WORLD

SD MANAGEMENT SUMMIT

SD 2003

WE'RE BUILDING THE TEAM THAT WILL BUILD OUR COMPANY

Sammy Studios is looking for talented and creative people to break new ground. Help lay the foundation for a new, advanced video game development and publishing house under construction by Sammy Corporation, one of Japan's most successful entertainment companies.

Positions are available in our office and studio complex in Los Angeles, California, and our state-of-the-art 47,000 sq. ft. facility located in the coastal city of Carlsbad in sunny San Diego County, California:

Producer
Lead Artist
Lead Programmer
Lead Game Designer
Senior Programmer
Senior Game Designer
Senior Artist
Artist
Associate Artist

Senior Concept Artist
Concept Artist
Level Layout Designer
Programmer
Associate Programmer
Assistant Producer
Associate Producer
Sound Engineer
Tester

Join veteran artists, designers, producers and programmers from leading video game and entertainment companies. Create original, franchise-worthy titles for the current generation of video game consoles. Help lay the groundwork for the company and grow as Sammy grows into a global video game developing and publishing force!

Established in 2002 by leading industry professionals, Sammy Studios is a global developer, publisher and distributor of video games and entertainment content for a variety of platforms, including Sony PlayStation® 2, Microsoft Xbox™, Nintendo Game Cube® and Game Boy® Advance, PC and Macintosh. The company operates offices in the US and Europe, with its headquarters and a state-of-the-art development studio located in Carlsbad, California, and a second office and studio complex located in Los Angeles, California. Sammy Studios is a wholly owned subsidiary of Sammy Corporation of Tokyo, Japan, an entertainment and amusement conglomerate with over a billion dollars in annual revenue and one of the top-rated companies traded on the First Section of the Tokyo Stock Exchange. For more information on the company, please visit www.sammystudios.com <http://www.sammystudios.com/> .

(Up-to-date listings and additional information are available at www.sammystudios.com <http://www.sammystudios.com/careers>)

Sammy Studios

TIMEGATE STUDIOS

The studio behind the award-winning ***Kohan: Immortal Sovereigns*** (2001 Strategy Game of the Year from *Computer Games Magazine, Computer Gaming World, and PC Gamer*) is hiring new team members to join its talented crew.

ART
3D Modelers, Animators, Texture Artists

DESIGN
Sr. and Jr. Designers

PROGRAMMING
Lead, 3D Graphics, A.I.

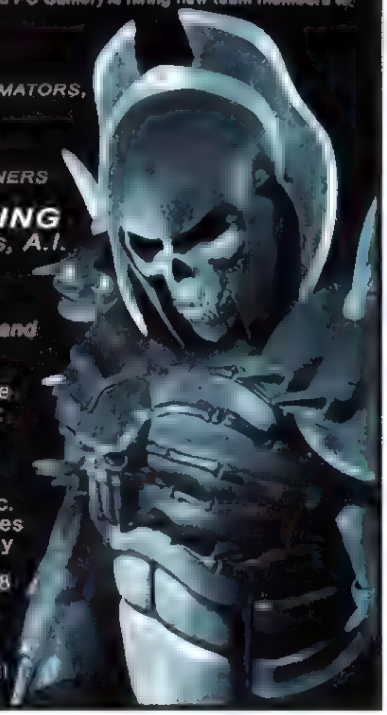
Exciting projects, competitive salaries, and excellent benefits!

To apply, send resume and portfolio/demo to:

jobs@timegate.com

TimeGate Studios, Inc.
Attn: Human Resources
14140 Southwest Frwy
Suite 400
Sugar Land, TX 77478

http://www.timegate.com



OPUS.CO.JP NEEDS

■ PROGRAMMERS ■ 3D ARTISTS ■ DESIGNERS

www.opus.co.jp

UFC THROWDOWN
© 2002 Crave Entertainment, Inc. All rights reserved. Published by Crave Entertainment, Inc. under license from Zuffa, LLC. Ultimate Fighting Championship, UFC and the UFC Logo are either trademarks or registered trademarks of Zuffa, LLC in the US and other countries and are used under license. Crave Entertainment and the Crave Entertainment logo are either trademarks or registered trademarks of Crave Entertainment, Inc. in the U.S. and other countries. All other trademarks and copyrights are the property of their respective owners.

Please send your resume to:
jobs@opus.co.jp
Inoue Bld. 5F, 3-14-12 Kamiosaki, Shinagawa-ku, 141-0021 Japan
+81-3-5447-1775-phone +81-3-5447-1770-fax

Are you Ready for Life on the eDGE?

Edge of Reality, Ltd. is a veteran entertainment software developer focused on next generation consoles. Located in beautiful Austin TX, the studio is currently hiring for a major license based project. Edge of Reality is currently looking for talented individuals with a passion for games to join our highly creative team!

eDGE OF rEALITY

Positions Available

- Senior Environment Artist
- Senior Character Artist
- Environment Artist
- Senior Animator
- Senior Tools programmer
- Effects/Lighting artist
- Senior Texture Artist
- Lead Designer

See our Web site for more details! www.edgeofreality.com

TAKING GAMES TO THE eDGE

Multidemensional Software Engineer (Boulder, CO)

Resp. include: analyze, design, program and modify graphic tools for console video games; responsible for conception, production, testing, debugging, delivery, documentation and maintenance of 3-D graphics tools; write code using current program languages and technologies; act as technical mentor to other engineers; work collaboratively with artists, animators, musicians and producers to achieve engineering goals.

Requirements: B.A. or B.S. in Computer Science, Engineering, Mathematics, Physics or equivalents, plus two years of experience in the job offered.

40 hrs. per week
Salary: $73,600 per year.
8:30 a.m. to 5:30 p.m., M-F.

The applicant must have proof of eligibility to work in the US. Mail resumes to Employment Programs, PO Box 46547, Denver, CO 80202, and refer to order number: **C05033064**.

Computer Graphics

- 2D & 3D Animation
- 3D Modeling
- Character Design
- Computer Arts
- Computer Illustration
- Digital Imaging
- Digital Media
- Digital Photography
- Gaming
- Visual Effects
- Web Design
- And More...

AA | BFA | MFA Degrees | Portfolio Development | Online Classes

Apply Now for Summer, Fall & Spring Semesters

High School Scholarships & Teacher Grants Available

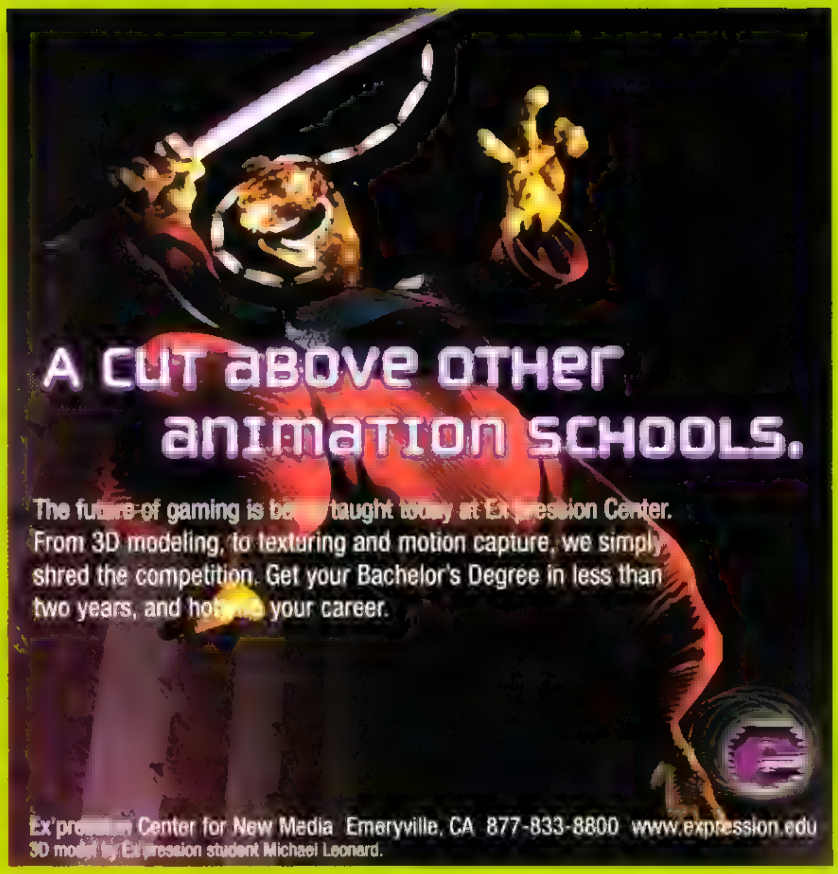
AcademyOfArtCollege
1.800.544.ARTS | www.academyart.edu

79 New Montgomery St. | San Francisco, CA 94105 | Nationally Accredited by ACICS, NASAD & FIDER | Established in 1929

THINK YOU COULD ROCK THE TECH WORLD?

Prove it.

UAT


A CUT ABOVE OTHER ANIMATION SCHOOLS.
The future of gaming is being taught today at Expression Center. From 3D modeling, to texturing and motion capture, we simply shred the competition. Get your Bachelor's Degree in less than two years, and hone your career.
Expression Center for New Media Emeryville, CA 877-833-8800 www.expression.edu
3D model by Expression student Michael Leonard.

Ai Center for Digital Imaging and Sound™

Game Development

"Small classes, passionate students – this is a prime environment for shaping the minds and skill-sets of tomorrow's interactive storytellers"

"I've played games all my life. Now I'm using software like Softimage™ and UnrealEd™ to make a real game."

Experience the difference.

Call for information about our educational programs offered

1-800-661-1885

artschool.com

3264 Beta Avenue, Burnaby, BC Canada V5G 4K4

...we make models
...we make textures
...we make animation

NEW PENCIL
WE MAKE ART

80 Liberty Ship Way, Suite 6 Sausalito, CA 94965
phone 415.339.1800 fax 415.339.1803
web www.newpencil.com

FULL SAIL
Real World Education

School of:
Game Design
Computer Animation
Digital Media
Film
Audio
Show Production

800.226.7625
www.fullsail.com

3300 University Boulevard
Winter Park, FL 32792

- Financial aid available to those who qualify.
- Job placement assistance.
- Accredited by ACCSCT.

©2001 Full Sail, Inc. All rights reserved.
The terms "Full Sail," "Full Sail Real World Education," and the Full Sail logo are either registered service marks or service marks of Full Sail, Inc.

If you're going to pay for an orange...

you should get an orange.

If you squeeze the new DVD from VFS you will never again be confused about who's got the juice.

To get more information or your own DVD of VFS student work call 1-800-661-4101 or email dvd@vfs.com

VFS Vancouver Film School
Creative. Disciplined. Focused.

It's Alive!
Terry Ziegelman
Paul George
Stephen Johnson
Jamie Kirschenbaum

BACHELOR OF FINE ARTS
MASTER OF ARCHITECTURE
MASTER OF ARTS
MASTER OF FINE ARTS

Savannah College of Art and Design

AN INTERNATIONAL UNIVERSITY FOR THE ARTS

Savannah, Georgia
800.869.7223 | www.scad.edu

·MUSIC
·SPEECH ·SOUND
·AUDIO INTEGRATION

"The best audio team I have ever worked with, in or out of house"
Michael Waite Producer Electronic Arts

"Omni Interactive Audio have proven over the years to be inventive highly professional and highly passionate in their craft. In spite of the odds they routinely turn out best in class audio transforming a good product into a AAA experience. Proactive great communicators and always looking to learn (and apply) new tricks to keep their skills at the front of the pack – highly recommended."
John Eberhardt Senior Program Manger, Microsoft

www.OmniInteractiveAudio.com

NPOWER

NEW 3DSMAX PLUG-INS!

Better Tools to Save You Time & Money!

Empowering Creativity with Advanced Technologies

POWER BOOLEANS

Fewer Polygons & Cleaner Meshes.
Fully Animatable Boolean Operations!
Improved Usability with Multi-Object Selection.
Integrated Decimation Polygon Reduction!

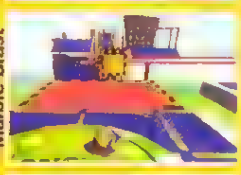
POWER SOLIDS

View Dependent Tessellation for Rendering.
Import IGES, Rhino,STEP & SAT files.
Powerful Construction Primatives.
Filleting, Shelling & Booleans!
Animation of all Solids Operations.

www.npowersoftware.com
(858)-538-3800

Invest in Your Own Game!

Realm Wars


Marble Blast

Metal Drift

- Award-Winning Network Code
- Engine and Tool Source
- Supports PC, Mac and Linux Platforms
- Seamless Indoor & Outdoor Render Engine
- Powerful Scripting Language

$100 per programmer seat for Indie developers. Commercial licenses available with NO royalties. The **Torque Game Engine** is supported and expanded by one of the world's largest communities of independent developers.

TORQUE GAME ENGINE

GarageGames.com INDEPENDENT GAMES

GarageGames

www.garagegames.com

345 Mill Street, Suite #200 • Eugene, OR 97401 • 541.345.3048

©2002, GarageGames, Inc. Torque Game Engine, Realm Wars and GarageGames are trademarks of GarageGames, Inc. Marble Blast is a trademark of Monster StudGos, Inc. Metal Drift is a trademark of Oils Interactive.

we hear it

sound design
music production
voice talent

SingularityStudios

i2i ANIMATION

ART, STAGE & TECHNOLOGY... EYE TO EYE

CONCEPT

RIGGING

MODELING

ANIMATION

MAYA • 3DMAX • FILMBOX

i2i ANIMATION, INC.

(503) 368-3067

WWW.I2IANIMATION.COM

Bring True Hollywood Sound To Your Game

SONICPOOL

Complete Gaming Sound

Specializing in 5.1 Dolby Surround & Sound Design

Sound Design/Voice Production
Trailers/Promos/Commercials
Episodic Television/Feature Film
Radio Commericals/Station ID's
Documentary/Corporate/Internet
Complete Music Production
Award Winning Mixers & Editors

DVD Commentary Recording
M&E Restoration
De-Noising/Restoration
ADR/Voice Over
ISDN/Phone Patches
Lounge/Kitchen/Broadband Internet Access
Small Run Multi-format Dubbing

6860 Lexington Ave Los Angeles, CA 90038
p 323.460.4649 f 323-460-6063
i www.sonicpool.com

Gamasutra.com

It's simple:

1. Search thousands of resumes by profession, game credits, technical skills, or preferred employment status.

2. Find experienced, professional game developers with **an average of 5 years of industry experience** and **more than 4 titles developed.**

3. Perform unlimited searches during your access period.

GamaNetwork

How to hire a game developer.

www.gamasutra.com/resumes

continued from page 64

have simulated personalities complete with nonverbal communication and speech recognition, we at least need more robust AIs that can make realistic personal decisions in a wide variety of scenarios beyond simple fight-or-flight.

Interface innovation. Button and thumbstick controllers are especially good for fighting things but not so great at other interactions with anthropomorphic characters. Interface innovations in speech and face/gesture recognition, and the use of input methods such as music and text, all possess untapped potential to elevate and vary our emotional experiences in games.

Emotional buy-in. What makes us care? The size of the stakes? Not necessarily. What's really important is how much a given situation or problem relates to us personally. Everyone agrees that if "the universe" or "good" as we know it is destroyed, the situation would certainly be personally relevant. But smaller stakes can seem equally important if they are made sufficiently personal. One way to achieve this personal relevance is through a direct responsibility for another single creature. SEAMAN and earlier, simpler sim creatures were examples of this, and other recent games have successfully used this formula, such as BLACK & WHITE and ICO. More importantly, personalizing the responsibility means that player responsibility becomes a facet of gameplay. In ICO you must actively keep Princess Yorda with you by pulling or coaxing her forward, and defend her from the constant threat of capture.

Design for player expression. As world builders, we can give the players tools and settings to inject themselves into the story, supporting a broader freedom within the constraints of the fictional setting. In GRAND THEFT AUTO 3, the best example to date of this kind of richly developed game world, players interact with the game world to write their own short story, while playing out the larger narrative. More often than not, it's those player-written stories that make the game memorable and generate excitement. That's why many players spurned GTA3's story mode entirely in favor of the open-ended play mode. To make better narrative games, we need to incorporate the lessons of systemic gameplay and integrate that play style into fiction arcs that aren't so easily discarded.

Development processes that encourage innovation. Games with characters and narrative are almost universally expensive. It's no coincidence so many indie games are puzzle games, card games, and space flight sims. Creating living, walking, talking beings is incredibly labor intensive. Large teams can saddle such risk because the potential rewards are large enough and because the genre is inherently appealing to established developers. Publishers in turn like the fact that recognizable characters help sell games. But indie developers don't have the same resources and infrastructure to make revolutionary narrative games. Mods offer some potential, but availability of better and cheaper tools for enabling deep character development will lead to more innovation, sooner.

For years we've survived on easy interactivity, and what I'm proposing here is not the easy stuff. But our medium will not expand until we make more effort to tackle the hard and messy problems. Not all games made with more complicated technology are going to be masterpieces of the form — in fact they most certainly won't be, initially. The biggest groundbreakers should try first to entertain. Mass appeal nabs publishers and others to fund projects; risk can be incremental, and amortized. Little by little, our creative and artistic landscape will grow; only then will our *Anna Kareninas* and *Citizen Kanes* emerge.

HEATHER KELLEY | *Heather is a designer on THIEF 3 at Ion Storm Austin. She has a master's degree in radio-TV-film from the University of Texas and has contributed production and design work to three published titles during her first five years in the industry. You can excoriate her at hkelley@ionstorm.com.*

ADVERTISER INDEX

COMPANY NAME	PAGE	COMPANY NAME	PAGE	COMPANY NAME	PAGE
3Dlabs	3	i2i Animation	61	RAD Game Tools	C4
Academy of Art College	57	Integrity Ware	60	Sammy Studios	55
ATI Technologies	33	Colorado Dept. of Labor	56	Savannah College of Art and Design	60
Butterfly.net	15	Metrowerks	29	Singularity Studios	61
Center for Digital Expression	58	Microsoft	21	Sonicpool	62
Criterion Software	C2, 19	New Pencil	59	Sony Securom	40
Discreet	C3	Nokia	8	Time Gate Studios	56
Edge of Reality	56	Omni Interactive Audio	62	University of Advanced Computer Technology	59
Expression Center for New Media	58	Perforce	7	Vancouver Film School	59
Full Sail Real World Education	59	Premier Press	27	Zona Inc.	23
Garage Games	61	Programmer's Paradise	13		
Havok	5	Qualcomm	11		

Narrative Games: Finding Another Side to the Story

As developers, we place a disproportionate stake of our medium's identity in character-based narrative games. These games give us something that we can't get from the intellectual beta-wave exercise of puzzle games or the adrenaline-pumping simulation of sports games. While puzzle and sports games represent a hefty chunk of game sales and game players, you seldom see a polygonal quarterback, much less a falling brick, on a magazine cover. We want to reach players emotionally in the best way we know how; because stories hold a privileged place in our culture — in all cultures, in fact — we use these games to represent our craft to the outside world. As humans, we use stories to make sense of the world, and narrative games are how we achieve that goal in our medium.

If we don't make such games relevant to wider audiences by increasing their variety and complexity, narrative games will ultimately hamper the evolution of games themselves. We must expand beyond the classic hero story to get these games past the status of geek recreation. It's not going to be easy, or pretty, but it's what we'll have to do if we want the game industry to grow up and achieve mainstream recognition as a form of art and human expression.

Console publishers have long known that the majority of the potential game-buying population doesn't give a damn that your game has dynamic LOD, bump-mapped decals, and pushes 1.5 trillion multi-pass lit polygons per second. If we want a larger audience, we need to shift some focus from the visual aspects of game technology and focus on innovations in player experiences.

To accomplish such a shift in narrative games, we need to innovate on a number of different fronts:

Narrative variety. Most story games are comparable to superhero comic books in terms of audience and themes. But why? Why not explore more varied subject matter? The majority of superhero comics and narrative games give us a chance to play with power. But that's not the only story there is to tell about being alive. We can innovate through our design and writing to explore other facets of existence; we can invent situations and stories that a player wants to experience, even if they don't fulfill these obvious power fantasies.

This kind of innovation poses a bigger challenge than it sounds. Hero narratives depend on the triumph of the protagonist. Since we embody the story's protagonist rather than simply watching the story unfold, such games leave no room for tragedy or failure. Failure can be an effective emotional experience as a bystander, but as long as we have choice, we do not naturally choose to fail. We don't mind empathizing with a tragic character if we see the machinations of the universe working his or her ruin, beyond our control, but we can't stand to be that person ourselves and be powerless to prevent it. So we need new story mechanics that either allow the player to fail and still be satisfied or provide other contexts for success that don't depend on an endless string of worlds-in-peril that only we can save.

Multiple fully realized characters. Most narrative games are plot driven, with no attention given to character subtlety. But this doesn't have to remain the case. Some games are already blazing compelling trails in this arena. For example, SEAMAN integrated the intimacy of a spoken interface with a nonheroic personal story arc. Player achievement in SEAMAN was on a much smaller scale but was as meaningful as saving the world. The day I accidentally killed my carefully nurtured two-month-old Seaman was probably the most emotionally charged moment I've had playing a game. On a technical level, until we

continued on page 63

Illustration by Steve Munday

3ds max

Every day, gamers buy more games created with **3ds max**™ than any other animation software. Which makes sense when you consider **3ds max** has helped create thousands of games in every genre and platform. Whether it's RPG, Strategy, Simulation or good clean fun plundering Glukkons, all **3ds max** games have one thing in common: Gamers can't seem to get enough.

discreet™

For more info on **3ds max** and other 3D products like **gmax**® and **character studio**®, call 1.800.869.3504 or go to www.discreet.com.

Discreet is a division of Autodesk, Inc. Discreet, character studio, 3ds max, and reactor are either registered trademarks or trademarks of Autodesk, Inc. in the U.S.A. and/or other countries. All other brand names, product names or trademarks belong to their respective holders. ©2002 Autodesk, Inc. All rights reserved. ©1987, 2001 Konami Computer Entertainment Japan, Planet Moon Studios, Oddworld: Munch's Oddysee ©2002 Oddworld Inhabitants, Inc. All Rights Reserved. The Oddworld logo, Oddworld: Munch's Oddysee are either registered trademarks or trademarks of Oddworld Inhabitants in the United States and/or other countries. Lionhead Studios/Electronic Arts™, Diablo® II box cover image courtesy of Blizzard Entertainment®, Remedy Entertainment & 3D Realms Entertainment, ©1999-2002 Activision, Inc. and its affiliates. Activision is a registered trademark and Activision 02, Tony Hawk's Pro Skater and Pro Skater are trademarks of Activision, Inc. and its affiliates. All Rights Reserved. Tony Hawk is a trademark of Tony Hawk. The ratings icon is a trademark of the Interactive Digital Software Association. Image courtesy of Electronic Arts™, 2002, ©Microsoft Corporation. All rights reserved. Microsoft, Bungie, Halo, Xbox and the Xbox logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Valve Software/Sierra.

The Best in Game Development Technology

Pixomatic Rendering Technology

Now you know what Michael Abrash and Mike Sartain have been up to! Designed for today's games, **Pixomatic** is their new software renderer that provides MIP mapping, bi-linear filtering, alpha-blending, alpha-test, point sprites, multi-texture, 32-bit and palettized textures, **DOT-3 bump mapping**, fog, specular, **stencil and z-buffering**, indexed primatives, multiple streams, high-speed blitting, and more! Pixomatic uses runtime code generation to create **optimized MMX, 3DNow!, and SSE** code on-the-fly! You will be amazed at its speed (what else would you expect from everyone's favorite optimizer?)! **Target the mass market** - don't let ancient 3D cards, buggy 3D drivers, or lack of 3D hardware prevent people from playing your games!

BINK VIDEO

Bink Video Technology

The very best video codec - make your videos shine! Unrivaled quality (**better than DVD** and MPEG II). Up to **three times faster** than other true-color codecs. Perceptibly lossless, **8 to 1 audio compression**. Decompresses to DirectDraw, DIBSections, YUV overlays, or any other memory. Available for Win32, MacOS, Xbox, and now **Nintendo GameCube**. Think Bink!

Granny 3D 2.2

Granny is a **sophisticated dynamic 3D animation system** with an optimized run-time engine that delivers incredible performance in a tiny memory footprint perfect for consoles, PCs and Macs. **MAX & Maya** plug-ins export materials, multiple vertex UVs and colors, mesh data, and complete animation and skinning information. Granny also now includes **amazing normal map generation** - give Granny a high and low-res model and she'll build the normal map! These features, plus advanced camera control, collision detection, animation and texture processing, MIP-map generation, and **Bink and S3TC** texture compression, all add years worth of features to your game in just a few hours of integration time!

MILES SOUND SYSTEM

Miles Sound System

The ultimate sound system for PC and Mac! Miles supports 2D and 3D audio, MIDI with DLS, streaming, CD audio, **DSP filtering**, **MP3 playback (patent rights included)**, **Internet voice chat** (2900, 2400, and 1200 bps codecs), Creative **EAX 1, 2 & 3**, Aureal **A3D 1 & 2**, RSX 3D, DirectX 7, Dolby Surround, QSound, **super-fast software EAX emulation**, and more!

RAD GAME TOOLS ©

425 893 4300

www.radgametools.com

Powerful Technology. Easy To Use. No Royalties.

Made with love by

Our goal is to preserve classic video game magazines so that they are not lost permanently.

People interested in helping out in any capacity, please visit us at retromags.com.

No profit is made from these scans, nor do we offer anything available from the publishers themselves.

If you come across anyone selling releases from this site, please do not support them and do let us know.

Thank you!